# 减缩轮 PRIDE 算法的线性分析

伊文坛,田 亚,陈少真
(数学工程与先进计算国家重点实验室,河南郑州 450001)

**摘 要:** PRIDE 是 Albrecht 等人在 2014 美密会上提出的轻量级分组密码算法. PRIDE 采用典型 SPN 密码结构,共迭代 20 轮. 其设计主要关注于线性层,兼顾了算法的效率和安全. 该文探讨了 $S$ 盒和线性层矩阵的线性性质,构造了 16 条优势为 $2^{-5}$ 的 2 轮线性逼近和 8 条优势为 $2^{-3}$ 的 1 轮线性逼近. 利用合适的线性逼近,结合密钥扩展算法、$S$ 盒的线性性质和部分和技术,我们对 18 轮和 19 轮 PRIDE 算法进行了线性分析. 该分析分别需要 $2^{60}$ 个已知明文,$2^{74.9}$ 次 18 轮加密和 $2^{62}$ 个已知明文,$2^{74.9}$ 次 19 轮加密. 另外,我们给出了一些关于 $S$ 盒差分性质和线性性质之间联系的结论,有助于减少攻击过程中的计算量. 本文是已知明文攻击. 本文是关于 PRIDE 算法的第一个线性分析.

**关键词:** 分组密码; PRIDE 算法; 线性分析; 线性逼近
**中图分类号:** TP309 **文献标识码:** A **文章编号:** 0372-2112 (2017)02-0468-09
**电子学报 URL:** http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2017.02.028

# Linear Cryptanalysis of Reduced-Round PRIDE Block Cipher

YI Wen-tan, TIAN Ya, CHEN Shao-zhen
(*State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, Henan* 450001, *China*)

**Abstract:** PRIDE is a light weight block cipher designed by Albrecht et al. in CRYPTO 2014, which adopts the classical SPN (Substitution Permutation Network) structure and iterates for 20 rounds. The construction of linear layers is very interesting and performances good both in security and efficiency. This paper investigates the properties of the S-boxes and the linear matrices, and then constructs 16 different 2-round iterative linear approximations with the bis $2^{-5}$ and 8 different 1-round iterative linear approximations with the bis $2^{-3}$. Base on some suitable approximations, attacks on 18-round and 19-round PRIDE are presented by means of linear cryptanalysis with the properties of key schedule, the linear characteristics and the partial-sum technique, which need about $2^{74.9}$ encryptions with $2^{60}$ known plaintexts and $2^{74.9}$ encryptions with $2^{62}$ known plaintexts, respectively. Furthermore, some interesting links between differential and linear characteristics are shown, which are helpful to reduce the compute complexity. Our analysis is the first linear attack on PRIDE block cipher with known plaintexts.

**Key words:** block cipher; PRIDE; linear cryptanalysis; linear approximation

## 1 引言

轻量级密码算法具有资源占用量较少的优点,特别适用于 RFID(Radio Frequency Identification)、无线传感技术等资源和计算能力有限的设备和环境中. 近年来,关于轻量级分组密码的研究越来越受到人们的关注,设计和公开了很多轻量级算法,比如 PRESENT[1], MIBS[2], LED[3], LBlock[4], SIMON 和 SPECK[5] 等算法. PRIDE[6] 是在 2014 年提出的一个轻量级分组密码算法. 其线性层的出色设计使得整个算法兼顾了效率和安全,在硬件和软件环境中都有优良的表现. 算法整体采用 SPN(Substitution Permutation Network)结构,分组长度为 64 比特,密钥长度为 128 比特,迭代 20 轮. 由于刚提出不久,目前针对 PRIDE 的安全性分析比较少,主要包括差分分析和相关密钥条件下的差分分析. 具体结果如下:

文献[7]利用 $S$ 盒和线性层的性质,构造了 16 条概率为 $2^{-8}$ 的 2 轮迭代的差分特征. 利用 15 轮的差分特征,对 18 轮 PRIDE 算法进行了分析. 该方法大体需要 $2^{60}$ 个选择明文和 $2^{66}$ 次 18 轮加密;该分析成功的概率

收稿日期:2015-07-15;修回日期:2016-01-04;责任编辑:蓝红杰
基金项目:信息保障技术重点实验室开放基金(No. KJ-13-010)

为 61%;

文献[8]改进了文献[7]中差分分析的结果;利用搜索到的概率为 $2^{-4}$ 的 1 轮迭代的差分特征,结合 $S$ 盒的一些差分性质,对 19 轮 PRIDE 算法进行了差分分析. 该分析需要的时间复杂度为 $2^{62}$ 次 19 轮加密,数据复杂度为 $2^{60}$ 个选择明文;

戴艺滨等[9]对 PRIDE 算法在相关密钥条件下抗差分分析的能力进行估计. 结果显示,在相关密钥条件下,RRIDE 算法存在概率为 $2^{-4}$ 的 2 轮迭代的差分特征,利用这些差分特征,对全轮 PRIDE 算法的分析需要 $2^{39}$ 个选择明文和 $2^{60}$ 次加密.

可以看出,对 PRIDE 算法的安全性评估只有差分分析,还没有关于线性分析[10,11]方面的结果. 线性分析是由 Matsui 在 1993 年提出的已知明文攻击方法,该方法利用分组密码算法输入与输出之间的优势较大的线性逼近来区分密码算法和随机函数,从而进行密钥恢复攻击的工作. 随后,线性分析方法不断的自我改进和完善,出现了多重线性分析[12,13],多维线性分析[14,15],以及最近提出的零相关线性分析[16~18]等. 同时,在很多算法中取得了重要的结果,比如 FEAL-8X[19],DES[10],PRESENT[20],MULTI2[21]等著名密码算法. 线性分析成为分组密码分析方法中有效的工具. 本文利用线性分析评估 PRIDE 密码算法的安全性. 主要结论有两点:

(1)对于 PRDIE 算法的 $S$ 盒运算,存在输入掩码和输出掩码相同,且重量为 1 的线性逼近,其优势为 $2^{-2}$. 而线性层包括 3 个运算:$64\times64$ 比特规模的置换矩阵 $\boldsymbol{P}$;4 个并置的 $16\times16$ 比特规模的变换矩阵 $\boldsymbol{L}_0,\boldsymbol{L}_1,\boldsymbol{L}_2,\boldsymbol{L}_3$,其中 $\boldsymbol{L}_0,\boldsymbol{L}_3$ 矩阵具有对角性质和对合性质,分支数为 3 的矩阵;$64\times64$ 比特规模的置换矩阵 $\boldsymbol{P}^{-1}$;利用这些性质,我们构造出 16 条优势为 $2^{-5}$ 的 2 轮线性逼近和 8 条优势为 $2^{-3}$ 的 1 轮线性逼近;给出了一些 $S$ 盒运算关于差分性质和线性性质之间联系的结论;利用 PRIDE 算法 $S$ 盒已有的一些差分性质,推导出一些线性性质. 这些差分性质和线性性质分别从不同的角度反映出 $S$ 盒设计的某些缺陷;

(2)利用线性分析方法,结合密钥扩展算法的特点和部分和技术,对 18 轮和 19 轮 PRIDE 密码算法做安全性评估;分析过程利用推导出的 $S$ 盒的线性性质减少了分析的计算复杂度. 此外,运用明文剖分技术也可以减少 10% 所需的已知明文量.

## 2　预备知识

### 2.1　一些记号

$\boldsymbol{a}\|\boldsymbol{b}$:向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 的连接;

$p,c$:明文和密文;

$\boldsymbol{M}^{\mathrm{T}},\boldsymbol{M}^{-1}$:矩阵 $\boldsymbol{M}$ 的转置矩阵和逆向矩阵;

$\boldsymbol{a}_{[i]}$:向量 $\boldsymbol{a}$ 的第 $i$ 比特,1 是最低位;

$\boldsymbol{a}[i,j,\cdots]$:向量 $\boldsymbol{a}$ 的 $i,j\cdots$ 比特值,$i>j$;

$\boldsymbol{a}[i,j,\cdots]$:向量 $\boldsymbol{a}$ 的第 $i,j\cdots$ 个半字节;

$\boldsymbol{K}_i[i,j,\cdots]$:第 $i$ 轮子密钥的第 $i,j\cdots$ 个半字节;

$I_i$:第 $i$ 轮函数的 64 比特输入;

$X_i$:第 $i$ 轮函数中密钥加运算的 64 比特输出;

$Y_i$:第 $i$ 轮函数中 $S$ 盒层运算的 64 比特输出;

$Z_i$:第 $i$ 轮函数中置换层运算的 64 比特输出;

$W_i$:第 $i$ 轮函数中线性层矩阵运算的 64 比特输出;

$O_i$:第 $i$ 轮函数 64 比特输出;

$\lfloor x\rfloor$:表示不超过 $x$ 的最大整数.

### 2.2　PRIDE 算法介绍

分组密码算法 PRIDE 总体采用了 SPN 密码结构,分组长度为 64 比特,并且迭代 20 轮. 轮函数 $F$ 由轮子密钥加、$S$ 盒变换、线性层组成. 为了保证加解密的相似性,最后一轮中轮函数省略了线性层,而采用一层密钥加运算来代替. 轮子密钥加运算是指每轮输入的 64 比特都与 64 比特长的轮子密钥相异或;$S$ 盒变换是由 16 个相同的规模为 $4\times4$ 的 $S$ 盒并置而成;其中 $S$ 盒具体参数见文献[6]. 线性层包括置换层 $\boldsymbol{P}$、矩阵变换层 $\boldsymbol{L}$ 和逆向置换 $\boldsymbol{P}^{-1}$,其中置换 $\boldsymbol{P}$ 可以表示为如下函数

$$y=p(x)=\{(x+3)\bmod 4\}\times16+\lfloor x/4\rfloor+1$$

矩阵变换层 $\boldsymbol{L}$ 包括 4 个并置的 $16\times16$ 比特规模的变换矩阵 $\boldsymbol{L}_0,\boldsymbol{L}_1,\boldsymbol{L}_2,\boldsymbol{L}_3$,其中当 $i=0,3$ 时

$$\boldsymbol{L}_i=\boldsymbol{L}_i^{-1}=\boldsymbol{L}_i^{\mathrm{T}}$$

则整个线性层可由矩阵 $\boldsymbol{P}$ 来表示,矩阵 $\boldsymbol{P}$ 表示为

$$\boldsymbol{P}=\boldsymbol{P}^{-1}\begin{pmatrix}\boldsymbol{L}_0&0&0&0\\0&\boldsymbol{L}_1&0&0\\0&0&\boldsymbol{L}_2&0\\0&0&0&\boldsymbol{L}_3\end{pmatrix}\boldsymbol{P}$$

其中,$\boldsymbol{P},\boldsymbol{P}^{-1},\boldsymbol{L}_0,\boldsymbol{L}_1,\boldsymbol{L}_2,\boldsymbol{L}_3$ 的具体矩阵表示、轮函数和整个算法流程图见参考文献[6].

PRIDE 算法密钥长度为 128 比特,前 64 比特作为前后白化密钥使用. 后 64 比特作为子密钥使用,每一轮改变其中的 32 比特.

令 $k$ 是 128 比特长种子密钥,并且

$$k=k_0\|k_1$$

其中,作为前 $k_0=k_2$ 后白化密钥使用,$k_1$ 作为成轮子密钥的种子密钥. 具体生成方式为

$$\begin{aligned}f_r(k_1)=&k_1[1,2]\|g_r^{(1)}(k_1[3,4])\|k_1[5,6]\\&\|g_r^{(2)}(k_1[7,8])\|k_1[9,10]\|g_r^{(3)}(k_1[11,12])\\&\|k_1[13,14]\|g_r^{(4)}(k_1[15,16]),\end{aligned}$$

对于 $i=1,2,3,4$,

$$g_r^{(i)}(x)=(x+a^i\cdot r)\bmod256;$$

其中,$a^i$ 分别取值为 193,165,81,197.

### 2.3 线性分析介绍

本小节主要介绍线性分析方法,首先给出线性逼近,相关系数和线性偏差的定义. 对于给定的 $n$ 比特长的输入和输出掩码 $\boldsymbol{\alpha},\boldsymbol{\beta}$ 以及 $F_2^n$ 上的函数 $f$,定义相应的线性逼近为

$$(\boldsymbol{\alpha}\xrightarrow{f}\boldsymbol{\beta});$$

简单记为($\boldsymbol{\alpha}\rightarrow\boldsymbol{\beta}$). 进一步,该线性逼近的相关系数则定义为

$$\begin{aligned}C_f(\boldsymbol{a},\boldsymbol{\beta}) &= \mathrm{Cor}_x(\boldsymbol{\beta}\cdot f(x)\oplus\boldsymbol{a}\cdot x)\\ &= 2\mathrm{Pr}_x(\boldsymbol{\beta}\cdot f(x)\oplus\boldsymbol{a}\cdot x=0)-1,\end{aligned}$$

同时,我们定义线性偏差,也称为线性优势为

$$\varepsilon=\frac{1}{2}|C_f(\boldsymbol{\alpha},\boldsymbol{\beta})|;$$

若 $\varepsilon=0$,则称该线性线性逼近是零相关线性逼近. 线性分析利用偏差较大的线性逼近来区分随机函数和密码算法,获取部分或者整个密钥信息. 给定偏差较大的线性逼近,利用 Matsuiś 算法 2[10] 可以进行密钥恢复. 给定一定量的明密文对,猜测添加轮部分的子密钥,将明密文对加解密至线性区分器的边界处,根据满足线性逼近的明密文对数的数量来区分正确密钥猜测与错误密钥猜测. 一般情况下假定正确密钥下的偏差要远大于错误密钥下的偏差,所以一般认为使统计数最大的密钥为候选密钥.

设 $E_r$ 是分组长度为 $n$ 比特,迭代轮数为 $r$ 的密码,密钥为 $k$,明文为 $p$,相应的密文 $c$ 表示为

$$c=E_r(p;k);$$

假设该算法存在一条 $r-1$ 轮偏差为 $\varepsilon$ 的线性逼近,并且将密文部分解密至线性逼近的边界处所涉及的轮子密钥为 $k$ 比特,线性分析过程如下:

**(1)数据收集阶段** 收集 $C_N\cdot\varepsilon^{-2}$ 个明文 $p$,经过加密得到相应的密文 $c$,$C_N$ 和成功概率有关.

**(2)分析阶段** 对于第 $r$ 轮中涉及到的 $2^k$ 个轮子密钥 $k_r$,并建立计数器 $N[k_r]$;

($a$)对于每一个明密文对($p$,$c$),将 $c$ 部分解密一轮 $F$,也就是计算

$$\bar{c}=F^{-1}(c;k_r);$$

($b$)计算该明密文对是否使得线性逼近成立,若成立,相应的计数器加 1;

**(3)密钥筛选阶段**

($a$)第二个阶段中计数器数值最大所对应的密钥认为是候选密钥;

($b$)利用已知明密文对,穷举和验证其他未猜测到的密钥信息.

分析过程将所有的密钥根据计数器数值大小进行排序,一般认为正确密钥对应的偏差是最大. 然而这并不一定正确. 所以,为提高成功概率,认为正确密钥出现在偏差最高的多个密钥范围内,进而下一步可以通过明密文对验证这个范围中的密钥以找出正确密钥. 假设在部分加解密中猜测 $k$ 比特的密钥,将偏差最高的 $r$ 个密钥作为候选密钥,用 Level-a 表示分析的优势,其中

$$a=k-\log_2 r;$$

若所使用的线性逼近的偏差为 $\varepsilon$. 然而在错误密钥下,该密码函数可以看作为随机函数,假设通过明密文对所统计的偏差趋于零. 由文献[22]可知,使用数量为 $N$ 对明密文进行 Level-a 优势的分析时,其成功概率为

$$P=\int_{-2\varepsilon\sqrt{N}+\Phi^{-1}(1-2^{-a-1})}^{+\infty}\varphi(x)\mathrm{d}x; \tag{1}$$

其中,$\Phi$ 和 $\varphi$ 分别表示标准正态分布函数与标准正态分布密度函数. 从公式可以看出,分析的成功概率与部分解密中涉及到密钥的数量 $k$ 有关.

## 3 可迭代的线性逼近

本小节主要介绍通过分析 $S$ 盒的性质和线性层中置换矩阵以及并置矩阵的性质构造关于 PRIDE 算法的可迭代的偏差最大的线性逼近. 关于 RRIDE 算法所采用的 $S$ 盒运算,存在输入输出掩码相同,并且重量为 1,偏差为 $2^{-2}$的线性逼近. 所谓向量的重量是指该向量中非零比特个数.

**命题 1** 对于 PRIDE 算法采用的 $S$ 盒运算,线性逼近

$$0\mathrm{x}1\cdot x\oplus 0\mathrm{x}1\cdot S(x)=0;$$

成立的概率为 0.75,偏差为 $2^{-2}$.

从线性层中的置换函数 $P$ 的定义可知,$P$ 将 $S$ 盒的最左面的第一比特映射到第一个矩阵块,将第二比特映射到第二个矩阵块,以此类推. 而对于第 4 个规模为 16×16 比特的矩阵 $\boldsymbol{L}_3$,则有

**命题 2** 对于 PRIDE 算法线性层中的第 4 个矩阵 $\boldsymbol{L}_3$,存在 8 条重量为 2 的 16 比特向量 $\boldsymbol{a}$,使得

$$\boldsymbol{L}_3\boldsymbol{a}=\boldsymbol{a};$$

其中 $\boldsymbol{a}$ 具体可以取值为

(1,0,0,0;0,0,0,0;1,0,0,0;0,0,0,0);
(0,1,0,0;0,0,0,0;0,1,0,0;0,0,0,0);
(0,0,1,0;0,0,0,0;0,0,1,0;0,0,0,0);
(0,0,0,1;0,0,0,0;0,0,0,1;0,0,0,0);
(0,0,0,0;1,0,0,0;0,0,0,0;1,0,0,0);
(0,0,0,0;0,1,0,0;0,0,0,0;0,1,0,0);
(0,0,0,0;0,0,1,0;0,0,0,0;0,0,1,0);
(0,0,0,0;0,0,0,1;0,0,0,0;0,0,0,1)

此外,存在 16 对重量分别为 1 和 3 的 16 比特向量 $\boldsymbol{\alpha},\boldsymbol{\beta}$ 使得

$$\boldsymbol{L}_3\boldsymbol{\alpha}=\boldsymbol{\beta},\boldsymbol{L}_3\boldsymbol{\beta}=\boldsymbol{\alpha}.$$

**命题 3**　对于 PRIDE 算法的线性层 $\boldsymbol{P}$,存在 8 个重量为 2 的 64 比特向量 $\boldsymbol{\alpha}$,使得

$$\boldsymbol{P}\boldsymbol{a}=\boldsymbol{a};$$

记 PRIDE 算法的非线性层运算为 $\bar{S}$,线性逼近

$$\boldsymbol{a}\cdot(x\oplus\bar{S}(x))=0,$$

的线性偏差为 $2^{-3}$;此外,存在 16 对重量分别为 1 和 3 的 64 比特向量 $\boldsymbol{\alpha},\boldsymbol{\beta}$,使得

$$\boldsymbol{P}\boldsymbol{\alpha}=\boldsymbol{\beta};\ \boldsymbol{P}\boldsymbol{\beta}=\boldsymbol{\alpha};$$

并且

$$\boldsymbol{a}\cdot(x\oplus\bar{S}(x))=0;\ \boldsymbol{\beta}\cdot(x\oplus\bar{S}(x))=0,$$

分别是偏差为 $2^{-2}$ 和 $2^{-4}$ 的线性逼近.

由命题 3 可以,对于 PRIDE 算法存在 8 条 1 轮迭代的偏差为 $2^{-3}$ 线性逼近和 16 条 2 轮迭代的偏差为 $2^{-5}$ 线性逼近,具体见表 1 和表 2.

**表 1　16 条偏差为 $2^{-5}$ 的 2 轮迭代的线性逼近**

| | |
|---|---|
| 1 | $(1,0,0,0;0,0,0,0;0,0,0,0;0,0,0,0)\xrightarrow{1\text{轮}}(1,0,0,0;1,0,0,0;0,0,0,0;1,0,0,0)\xrightarrow{1\text{轮}}(1,0,0,0;0,0,0,0;0,0,0,0;0,0,0,0)$ |
| 2 | $(0,1,0,0;0,0,0,0;0,0,0,0;0,0,0,0)\xrightarrow{1\text{轮}}(0,1,0,0;0,1,0,0;0,0,0,0;0,1,0,0)\xrightarrow{1\text{轮}}(0,1,0,0;0,0,0,0;0,0,0,0;0,0,0,0)$ |
| 3 | $(0,0,1,0;0,0,0,0;0,0,0,0;0,0,0,0)\xrightarrow{1\text{轮}}(0,0,1,0;0,0,1,0;0,0,0,0;0,0,1,0)\xrightarrow{1\text{轮}}(0,0,1,0;0,0,0,0;0,0,0,0;0,0,0,0)$ |
| 4 | $(0,0,0,1;0,0,0,0;0,0,0,0;0,0,0,0)\xrightarrow{1\text{轮}}(0,0,0,1;0,0,0,1;0,0,0,0;0,0,0,1)\xrightarrow{1\text{轮}}(0,0,0,1;0,0,0,0;0,0,0,0;0,0,0,0)$ |
| 5 | $(0,0,0,0;1,0,0,0;0,0,0,0;0,0,0,0)\xrightarrow{1\text{轮}}(1,0,0,0;1,0,0,0;1,0,0,0;0,0,0,0)\xrightarrow{1\text{轮}}(0,0,0,0;1,0,0,0;0,0,0,0;0,0,0,0)$ |
| 6 | $(0,0,0,0;0,1,0,0;0,0,0,0;0,0,0,0)\xrightarrow{1\text{轮}}(0,1,0,0;0,1,0,0;0,1,0,0;0,0,0,0)\xrightarrow{1\text{轮}}(0,0,0,0;0,1,0,0;0,0,0,0;0,0,0,0)$ |
| 7 | $(0,0,0,0;0,0,1,0;0,0,0,0;0,0,0,0)\xrightarrow{1\text{轮}}(0,0,1,0;0,0,1,0;0,0,1,0;0,0,0,0)\xrightarrow{1\text{轮}}(0,0,0,0;0,0,1,0;0,0,0,0;0,0,0,0)$ |
| 8 | $(0,0,0,0;0,0,0,1;0,0,0,0;0,0,0,0)\xrightarrow{1\text{轮}}(0,0,0,1;0,0,0,1;0,0,0,1;0,0,0,0)\xrightarrow{1\text{轮}}(0,0,0,0;0,0,0,1;0,0,0,0;0,0,0,0)$ |
| 9 | $(0,0,0,0;0,0,0,0;1,0,0,0;0,0,0,0)\xrightarrow{1\text{轮}}(0,0,0,0;1,0,0,0;1,0,0,0;1,0,0,0)\xrightarrow{1\text{轮}}(0,0,0,0;0,0,0,0;1,0,0,0;0,0,0,0)$ |
| 10 | $(0,0,0,0;0,0,0,0;0,1,0,0;0,0,0,0)\xrightarrow{1\text{轮}}(0,0,0,0;0,1,0,0;0,1,0,0;0,1,0,0)\xrightarrow{1\text{轮}}(0,0,0,0;0,0,0,0;0,1,0,0;0,0,0,0)$ |
| 11 | $(0,0,0,0;0,0,0,0;0,0,1,0;0,0,0,0)\xrightarrow{1\text{轮}}(0,0,0,0;0,0,1,0;0,0,1,0;0,0,1,0)\xrightarrow{1\text{轮}}(0,0,0,0;0,0,0,0;0,0,1,0;0,0,0,0)$ |
| 12 | $(0,0,0,0;0,0,0,0;0,0,0,1;0,0,0,0)\xrightarrow{1\text{轮}}(0,0,0,0;0,0,0,1;0,0,0,1;0,0,0,1)\xrightarrow{1\text{轮}}(0,0,0,0;0,0,0,0;0,0,0,1;0,0,0,0)$ |
| 13 | $(0,0,0,0;0,0,0,0;0,0,0,0;1,0,0,0)\xrightarrow{1\text{轮}}(1,0,0,0;0,0,0,0;1,0,0,0;1,0,0,0)\xrightarrow{1\text{轮}}(0,0,0,0;0,0,0,0;0,0,0,0;1,0,0,0)$ |
| 14 | $(0,0,0,0;0,0,0,0;0,0,0,0;0,1,0,0)\xrightarrow{1\text{轮}}(0,1,0,0;0,0,0,0;0,1,0,0;0,1,0,0)\xrightarrow{1\text{轮}}(0,0,0,0;0,0,0,0;0,0,0,0;0,1,0,0)$ |
| 15 | $(0,0,0,0;0,0,0,0;0,0,0,0;0,0,1,0)\xrightarrow{1\text{轮}}(0,0,1,0;0,0,0,0;0,0,1,0;0,0,1,0)\xrightarrow{1\text{轮}}(0,0,0,0;0,0,0,0;0,0,0,0;0,0,1,0)$ |
| 16 | $(0,0,0,0;0,0,0,0;0,0,0,0;0,0,0,1)\xrightarrow{1\text{轮}}(0,0,0,1;0,0,0,0;0,0,0,1;0,0,0,1)\xrightarrow{1\text{轮}}(0,0,0,0;0,0,0,0;0,0,0,0;0,0,0,1)$ |

**表 2　8 条偏差为 $2^{-3}$ 的 1 轮迭代的**

| | |
|---|---|
| 1 | $(1,0,0,0;0,0,0,0;1,0,0,0;0,0,0,0)\xrightarrow{1\text{轮}}(1,0,0,0;0,0,0,0;1,0,0,0;0,0,0,0)$ |
| 2 | $(0,1,0,0;0,0,0,0;0,1,0,0;0,0,0,0)\xrightarrow{1\text{轮}}(0,1,0,0;0,0,0,0;0,1,0,0;0,0,0,0)$ |
| 3 | $(0,0,1,0;0,0,0,0;0,0,1,0;0,0,0,0)\xrightarrow{1\text{轮}}(0,0,1,0;0,0,0,0;0,0,1,0;0,0,0,0)$ |
| 4 | $(0,0,0,1;0,0,0,0;0,0,0,1;0,0,0,0)\xrightarrow{1\text{轮}}(0,0,0,1;0,0,0,0;0,0,0,1;0,0,0,0)$ |
| 5 | $(0,0,0,0;1,0,0,0;0,0,0,0;1,0,0,0)\xrightarrow{1\text{轮}}(0,0,0,0;1,0,0,0;0,0,0,0;1,0,0,0)$ |
| 6 | $(0,0,0,0;0,1,0,0;0,0,0,0;0,1,0,0)\xrightarrow{1\text{轮}}(0,1,0,0;0,0,0,0;0,1,0,0;0,1,0,0)$ |
| 7 | $(0,0,0,0;0,0,1,0;0,0,0,0;0,0,1,0)\xrightarrow{1\text{轮}}(0,0,1,0;0,0,0,0;0,0,1,0;0,0,1,0)$ |
| 8 | $(0,0,0,0;0,0,0,1;0,0,0,0;0,0,0,1)\xrightarrow{1\text{轮}}(0,0,0,1;0,0,0,0;0,0,0,1;0,0,0,1)$ |

## 4　$S$ 盒运算的线性性质

本小节主要讨论 PRIDE 算法 $S$ 盒运算的一些性质,这将有助于减少在线性分析中的计算复杂度. 首先,给出一个关于 $S$ 盒运算差分性质和线性性质联系的结论.

**定理 1**　对于 4×4 比特规模的 $S$ 盒运算,则下面两个条件等价:

($a$)截断差分特征

$$(1,0,0,0)\rightarrow(?,0,?,?);$$

概率为 1.

($b$)设线性逼近的输出掩码为 $\bar{\boldsymbol{\beta}}=(0,1,0,0)$. 若偏差非零,则输入掩码 $\bar{\boldsymbol{\alpha}}$ 的第 1 比特必为零,也就是 $\bar{\boldsymbol{\alpha}}$ 只能从集合

$$\{\boldsymbol{\alpha}\mid\boldsymbol{\alpha}=(0,?,?,?)\},$$

中取值,其中? 表示该比特位置未知.

由概率为 1 的差分特征,$S$ 盒输入的第 1 比特无论是 0 还是 1,对于输入的第 2 比特都相同. 输入的第 1 比特不影响输出的第 2 比特. 若线性逼近的输出掩码的第 2 比特为 1,其他比特为 0,线性偏差非零,输入掩

码的第 1 比特必然为 0. 文献[8]给出了关于 PRIDE 算法 $S$ 盒运算下面的差分性质.

**命题 4**[8] 对于 PRIDE 算法采用的 $S$ 盒运算,若差分特征的输入差分为(1,0,0,0),则输出差分具有形式(?,0,?,?);若输入差分为(0,0,0,1),则输出差分具有(0,?,?,?)的形式;由于 $S$ 盒运算具有对合性,也就是 $S^2=E$,反之,若差分特征的输出差分为(1,0,0,0),则输入差分具有形式(?,0,?,?);若差分特征的输出差分为(0,0,0,1),则输入差分具有形式(0,?,?,?);其中 $E$ 是单位映射.

从定理 1 和命题 4,得到关于 $S$ 盒的线性性质.

**推论 1** 对于 PRIDE 算法采用的 $S$ 盒运算,若偏差非零的线性逼近的输出掩码为(0,1,0,0),则输入差分具有形式(0,?,?,?);若偏差非零的线性逼近的输出掩码为(1,0,0,0),则输入掩码具有形式(?,?,?,0);反之,若输入掩码为(0,1,0,0),并且线性偏差非零,则输入掩码具有形式(0,?,?,?). 若输入掩码为(1,0,0,0),并且线性偏差非零,则输入掩码具有形式(?,?,?,0).

## 5 18 轮 PRIDE 算法的线性分析

本节主要利用上面构造的第 2 条 1 轮迭代的偏差为 $2^{-3}$ 的线性逼近迭代 14 轮作为区分器,往前扩展 2 轮并且往后扩展 2 轮,结合轮子密钥之间的关系和部分和技术,对 18 轮 PRIDE 算法做线性分析,见表 3. 另外,利用 $S$ 盒的线性性质可以减少分析过程中的计算量. 具体攻击过程如下:

**(1)数据收集阶段** 建立 8 比特的计数器 $N_0[x_0]$,且置为零. 取

$$x_0=X_1[2,6,9,10,14]\|X_1[3]_{[2,3,4]}\|X_1[13]_{[2,3,4]}\|Y_{18}[1,2,4,5,6,9,10,12,14]\|Y_{18}[7]_{[2,3,4]}\|Y_{18}[13]_{[2,3,4]}\|Y_{18}[16]_{[2,3,4]};$$

收集 $N$ 个明文密文对$(p,c)$,并计算

$$X_1=P^{-1}p;Y_{18}=P^{-1}c;$$

相应的计数器 $N_0[x_0]$ 加 1. 不超过 $2^{64}$ 个明密文对分成 $2^{75}$ 个状态,所以计数器 8 比特够用.

**(2)分析阶段** 建立计数器 $N[k]$,且置为零,

$$k=\{k_0\oplus f_1(k_1)\}[2,6,7,9,10,14]\|\{k_0\oplus f_1(k_1)\}[3,13]_{[2,3,4]}\|k_0[1,4,5,6,9,10,14]\|k_0[8,13,16]_{[2,3,4]}\|\{P^{-1}f_{18}(k_1)\}[2,10]$$

然后进行部分加解密过程:

(*a*)建立计数器 $N_1[x_1]$,且置为零. 取

$$x_1=X_1[3]_{[2,3,4]}\|X_1[7]\|Y_{18}[1,2,4,5,6,9,10,12,14]\|Y_{18}[7]_{[2,3,4]}\|Y_{18}[13]_{[2,3,4]}\|Y_{18}[16]_{[2,3,4]}\|I_2[2,10]$$

穷举 23 比特轮子密钥$\{k_0\oplus f_1(k_1)\}[2,6,9,10,14]$和$\{k_0\oplus f_1(k_1)\}[13]_{[2,3,4]}$,计算

$$Y_1[i]=S(X_1[i]\oplus\{k_0\oplus f_1(k_1)\}[i]);$$

其中 $i=2,6,9,10,14$;另外,有推论 1 可知,$S$ 盒的输出的第 2 比特仅与输入的后 3 比特有关,与第 1 比特无关,所以可以由 $X_1[13]\oplus f_1(k_1)\}[13]_{[2,3,4]}$ 计算出 $Y_1[13]_{[2]}$. 然后计算

$$I_2[10]_{[1]}=Y_1[6]_{[1]}\oplus Y_1[11]_{[1]}\oplus Y_1[13]_{[1]};$$
$$I_2[10]_{[2]}=Y_1[2]_{[2]}\oplus Y_1[13]_{[2]}\oplus Y_1[14]_{[2]};$$
$$I_2[10]_{[3]}=Y_1[6]_{[3]}\oplus Y_1[9]_{[3]}\oplus Y_1[10]_{[3]};$$
$$I_2[10]_{[4]}=Y_1[6]_{[4]}\oplus Y_1[10]_{[4]}\oplus Y_1[14]_{[4]};$$
$$I_2[2]_{[1]}=Y_1[6]_{[1]}\oplus Y_1[10]_{[1]}\oplus Y_1[13]_{[1]};$$
$$I_2[2]_{[2]}=Y_1[2]_{[2]}\oplus Y_1[13]_{[2]};$$
$$I_2[2]_{[3]}=Y_1[6]_{[3]}\oplus Y_1[9]_{[3]};$$
$$I_2[2]_{[4]}=Y_1[2]_{[4]}\oplus Y_1[6]_{[4]}\oplus Y_1[14]_{[4]};$$

然后更新计数器 $N_1[x_1]+=N_0[x_0]$. 此步需要 $N\times 2^{23}$ 次内存访问.

(*b*)建立计数器,且置为零. 取

$$x_2=X_1[3]_{[2,3,4]}\|Y_{18}[1,2,4,5,6,9,10,12,14]\|Y_{18}[8]_{[2,3,4]}\|Y_{18}[13]_{[2,3,4]}\|Y_{18}[16]_{[2,3,4]}\|I_2[2,10];$$

穷举 4 比特轮子密钥$\{k_0\oplus f_1(k_1)\}[7]$,计算并更新

**表 3 18 轮 PRIDE 算法的线性分析**

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $X_2$ | 0000 | ???? | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | ???? | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| $Y_2$ | 0000 | 0001 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0001 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| $Z_2$ | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0100 | 0000 | 0100 | 0000 |
| $W_2$ | 0000 | 0001 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0001 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| $I_3$ | 0000 | 0001 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0001 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| $X_{17}$ | 0000 | 0001 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0001 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| $Y_{17}$ | 0000 | ???? | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | ???? | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| $Z_{17}$ | 0? 00 | 0000 | 0? 00 | 0000 | 0? 00 | 0000 | 0? 00 | 0000 | 0? 00 | 0000 | 0? 00 | 0000 | 0? 00 | 0000 | 0? 00 | 0000 |
| $W_1$ | 0? 00 | 0? 00 | 0? 00 | 0? 00 | ? 000 | ? 00? | 0000 | ?? 0? | ? 00? | ? 000 | ?? 0? | 0000 | 0? 00 | 0? 00 | 0? 00 | 0? 00 |
| $I_{18}$ | 0?? 0 | ? 00? | 0000 | 00? 0 | 0?? 0 | ? 00? | 0000 | 0? 00 | 00? 0 | ? 0?? | 0000 | 00? 0 | 0? 00 | ?? 0? | 0000 | 0? 00 |
| $X_{18}$ | 0?? 0 | ? 00? | 0000 | 00? 0 | 0?? 0 | ? 00? | 0000 | 0? 00 | 00? 0 | ? 0?? | 0000 | 00? 0 | 0? 00 | ?? 0? | 0000 | 0? 00 |
| $Y_{18}$ | ???? | ???? | 0000 | ???? | ???? | ???? | 0000 | 0??? | ???? | ???? | 0000 | ???? | 0??? | ???? | 0000 | 0??? |
| $O_1$ | ???? | ???? | 0000 | ???? | ???? | ???? | 0000 | 0??? | ???? | ???? | 0000 | ???? | 0??? | ???? | 0000 | 0??? |

$$Y_1[7]=S(X_1[7]\oplus\{k_0\oplus f_1(k_1)\}[7]);$$

$$I_2[10]_{[3]}=I_2[10]_{[3]}\oplus Y_1[7]_{[3]};$$

然后更新计数器 $N_2[x_2]+=N_1[x_1]$. 此步需要 $2^{60}\times 2^{23}\times 2^4$ 次内存访问.

(c)建立计数器 $N_3[x_3]$,且置为零. 取

$$x_3=X_{18}[1,2,4,5,6,9,10,12,14]\parallel Y_{18}[8]_{[2,3,4]}\parallel Y_{18}[13]_{[2,3,4]}\parallel Y_{18}[16]_{[2,3,4]}\parallel I_2[2,10];$$

穷举 3 比特密钥 $\{k_0\oplus f_1(k_1)\}[3]_{[2,3,4]}$,计算 $Y_1[3]_{[2]}$ 更新

$$I_2[2]_{[2]}=I_2[2]_{[2]}\oplus Y_1[3]_{[2]};$$

然后更新计数器 $N_3[x_3]+=N_2[x_2]$. 此步大体需要 $2^{56}\times 2^{27}\times 2^3$ 次内存访问.

(d)建立计数器 $N_4[x_4]$,且置为零. 取

$$x_4=Y_{18}[1,4,5,6,9,10,12,14]\parallel Y_{18}[8]_{[2,3,4]}\parallel Y_{18}[13]_{[2,3,4]}\parallel Y_{18}[16]_{[2,3,4]}\parallel I_2[10]\parallel I_3[2]_{[4]}\parallel Y_{17}[2]_{[4]}\parallel Y_{17}[10]_{[1]};$$

穷举 4 比特轮子密钥 $f_2(k_1)[2]$,且计算 $k_0[2]$;计算

$$I_3[2]_{[4]}=S(I_1[2]\oplus f_2(k_1)[2])_{[4]};$$

$$Y_{17}[2]_{[4]}=S(Y_{18}[2]\oplus k_0[2])_{[4]};$$

$$Y_{17}[10]_{[1]}=S(Y_{18}[2]\oplus k_0[2])_{[1]};$$

然后更新计数器 $N_4[x_4]+=N_3[x_3]$. 此步大体需要 $2^{53}\times 2^{30}\times 2^4$ 次内存访问.

(e)建立计数器 $N_5[x_5]$,且置为零. 取

$$x_5=Y_{18}[1,4,5,6,9,12,14]\parallel Y_{18}[8]_{[2,3,4]}\parallel Y_{18}[13]_{[2,3,4]}\parallel Y_{18}[16]_{[2,3,4]}\parallel I_3[2]_{[4]}\parallel Y_{17}[2]_{[1,4]}\parallel Y_{17}[10]_{[1,3,4]};$$

穷举 4 比特轮子密钥 $f_2(k_1)[10]$,得到 $k_0[10]$ 且计算

$$I_3[2]_{[4]}=I_3[2]_{[4]}\oplus S(I_1[10]\oplus f_2(k_1)[10])_{[4]};$$

$$Y_{17}[2]_{[1]}=S(Y_{18}[10]\oplus k_0[10])_{[1]};$$

$$Y_{17}[10]_{[3]}=S(Y_{18}[10]\oplus k_0[10])_{[3]};$$

$$Y_{17}[10]_{[4]}=S(Y_{18}[10]\oplus k_0[10])_{[4]},$$

然后更新计数器 $N_5[x_5]+=N_4[x_4]$. 此步大体需要 $2^{48}\times 2^{34}\times 2^4$ 次内存访问.

(f)依次穷举密钥 $k_0[9]$,$k_0[6]$. 最后,计算并更新 $N_6[x_6]$,其中

$$x_6=Y_{18}[1,4,5,12,14]\parallel Y_{18}[8]_{[2,3,4]}\parallel Y_{18}[13]_{[2,3,4]}\parallel Y_{18}[16]_{[2,3,4]}\parallel I_3[2]_{[4]}\parallel Y_{17}[2]_{[1,4]}\parallel Y_{17}[10]_{[1,3,4]};$$

此步计算量不超过 $2\times 2^{85}$ 次内存访问.

(g)依次穷举密钥 $k_0[8]_{[2,3,4]}$,$k_0[16]_{[2,3,4]}$,$k_0[1]$. 计算并更新 $N_7[x_7]$,其中

$$x_7=Y_{18}[4,5,12,14]\parallel Y_{18}[13]_{[2,3,4]}\parallel I_3[2]_{[4]}\parallel Y_{17}[2]_{[1,2,4]}\parallel Y_{17}[10]_{[1,3,4]};$$

此步大体需要不超过 $7\times 2^{85}$ 次的内存访问.

(h)依次穷举密钥 $k_0[4]$,$k_0[12]$. 计算并更新 $N_8[x_8]$,其中

$$x_8=Y_{18}[5]\parallel Y_{18}[13]_{[2,3,4]}\parallel I_3[2]_{[4]}\parallel Y_{17}[2]_{[1,2,3,4]}\parallel Y_{17}[10]_{[1,3,4]};$$

此步大体需要不超过 $3\times 2^{87}$ 次的内存访问.

(i)依次穷举密钥 $k_0[13]_{[2,3,4]}$,$k_0[4]$. 计算并更新 $N_9[x_9]$,其中

$$x_8=I_3[2]_{[4]}\parallel Y_{17}[2]_{[1,2,3,4]}\parallel Y_{17}[10]_{[1,2,3,4]};$$

此步大体需要不超过 $5\times 2^{87}$ 次的内存访问.

(j)依次穷举密钥 $\{P^{-1}f_{18}(k_1)\}[2]$,$\{P^{-1}f_{18}(k_1)\}[10]$. 计算并更新 $N_{10}[x_{10}]$,其中

$$x_{10}=I_3[2]_{[4]}$$

此步大体需要不超过 $2^{89}$ 次的内存访问.

**(3)密钥筛选阶段**　经过部分加解密运算,得到线性逼近的临界状态. 若 $x_{10}=0$,则令 $N[k]+=N_{10}[x_{10}]$. 若 $x_{10}=1$,则令 $N[k]-=N_{10}[x_{10}]$. 将 $2^{67}$ 个绝对值较高的密钥作为候选密钥,也就是做优势为 Level-8 的分析,保留计数绝对值器最高的 $2^{67}$ 个密钥.

**复杂度估计**　在攻击过程中,我们选择 $N=4\times\varepsilon^{-2}=2^{60}$ 个已知明密文对. 在分析阶段中,第(a)步需要 $2^{60}\times 2^{23}$ 次内存访问. 第(b)步需要 $2^{60}\times 2^{23}\times 2^4$ 次内存访问;第(c)步需要 $2^{56}\times 2^{27}\times 2^3$ 次内存访问;第(d)步需要 $2^{53}\times 2^{30}\times 2^4$ 次内存访问;第(e)步需要 $2^{48}\times 2^{34}\times 2^4$ 次内存访问;第(f)~(i)步需要不超过 $57\times 2^{85}$ 次的内存访问;若访问一次内存等同于 1 轮加密,则一共需要 $2^{87}$ 次 18 轮加密. 分析过程中一共涉及 83 比特密钥,所以存储复杂度为 $2^{80}$ 字节. 由文献[22]可知,成功概率为 87%.

将 2 轮迭代的线性逼近迭代 7.5 轮可以得到偏差 $2^{-30}$ 的 15 轮线性逼近作为线性区分器. 以第 5 条 2 迭代的线性逼近为例,在 15 轮的基础上,往前扩展 2 轮,往后扩展 1 轮,对 18 轮的 PRIDE 算法做线性分析,需要 $2^{62}$ 个已知明密文对,需要时间复杂度为 $2^{60}$ 次 18 轮加密.

另外,对于 PRIDE 算法的 $S$ 盒运算

$$(y_1,y_2,y_3,y_4)=S(x_1,x_2,x_3,x_4);$$

在 $x_3=1$ 的条件下,线性逼近

$$y_4\oplus x_4=0,$$

成立的概率为 7/8,偏差为 3/8,比整个空间上的线性逼近的偏差 1/4 有所增大. 应用文献[19]中明文剖分的思想,可以减少攻击过程中的已知明文量,约为原来的 0.89 倍,但是计算量会略微增大.

## 6　19 轮 PRIDE 算法的线性分析

本节主要利用上构造的第 5 条 2 轮迭代的偏差为 $2^{-5}$ 的线性逼近迭代 7.5 次,形成 15 轮偏差为 $2^{-30}$ 的线性逼近作为区分器,往前往后各扩展 2 轮,结合轮子密钥之间的关系和部分和技术,对 19 轮 PRIDE 算法做线

性分析,见表4. 攻击过程如下:

**(1)数据收集阶段** 建立8比特的计数器 $V_0[y_0]$,且置为零. 取

$$y_0 = X_1[1,2,5,6,9,12,13,16] \parallel Y_{19}[1,3,4,5,7,8,9,11,13,15,16] \parallel Y_{19}[12]_{[2,3,4]}$$

收集 $N$ 个明文密文对$(m,c)$,并计算

$$X_1 = p^{-1}m; Y_{19} = p^{-1}c,$$

相应的计数器 $N_0[y_0]$ 加1. 不超过 $2^{64}$ 个明密文对分成 $2^{79}$ 个状态,所以8比特计数器合适.

**(2)分析阶段** 建立计数器 $V[k]$,且置为0,其中

$$k = \{k_0 \oplus f_1(k_1)\}[1,2,5,6,9,12,13,16] \parallel k_0[3,4,5,7,8,9,11,13,15,16] \parallel k_0[12]_{[2,3,4]} \parallel \{P^{-1}f_{19}(k_1)\}[1,5,9];$$

然后进行部分加解密过程:

(a)建立计数器 $V_1[y_1]$,且置为零. 取

$$y_1 = Y_{19}[3,4,5,7,8,9,11,13,15,16] \parallel Y_{19}[12]_{[2,3,4]} \parallel I_2[5,8] \parallel Y_{18}[1]_{[4]} \parallel Y_{18}[5]_{[1,4]} \parallel Y_{18}[8]_{[1]} \parallel I_3[5]_{[4]};$$

穷举36比特轮子密钥 $\{k_0 \oplus f_1(k_1)\}[1,2,5,6,9,12,13,16]$,以及 $f_2(k_1)[1]$,得到 $k_0[1]$,计算

$$Y_1[i] = S(X_1[i] \oplus \{k_0 \oplus f_1(k_1)\}[i]);$$

其中 $i = 1,2,5,6,9,12,13,16$;

$$X_{19}[1] = S(Y_{19}[1] \oplus k_0[1]);$$

然后经过部分加解密运算,得到

$I_2[1,5,9]$; $I_3[5]_{[4]} = S(I_3[1] \oplus f_2(k_1)[1])_{[4]}$;

$Y_{18}[1]_{[4]} = X_{19}[1]_{[4]}$; $Y_{18}[5]_{[1,4]} = X_{19}[1]_{[1,4]}$;

$Y_{18}[9]_{[1]} = X_{19}[1]_{[1]}$;

更新 $V_1[y_1] += V_0[y_0]$. 此步需要 $N \times 2^{36}$ 次内存访问.

(b)建立计数器 $V_2[y_2]$,且置为零. 取

$$y_2 = Y_{19}[3,4,7,8,9,11,13,15,16] \parallel Y_{19}[12]_{[2,3,4]} \parallel I_2[8] \parallel Y_{18}[1]_{[1,4]} \parallel Y_{18}[5]_{[1,4]} \parallel Y_{18}[8]_{[1,4]} \parallel I_3[5]_{[4]}$$

穷举4比特轮子密钥 $f_2(k_1)[5]$,得到 $k_0[5]$ 计算和更新

$I_3[5]_{[4]} = I_3[5]_{[4]} \oplus S(I_3[5] \oplus f_2(k_1)[5])_{[4]}$;

$X_{19}[5] = S(Y_{19}[5] \oplus k_0[5])$;

$Y_{18}[1]_{[1]} = X_{19}[1]_{[1]}$;

$Y_{18}[1]_{[4]} = Y_{18}[1]_{[4]} \oplus X_{19}[1]_{[4]}$;

$Y_{18}[5]_{[1,4]} = X_{19}[1]_{[1,4]}$;

$Y_{18}[9]_{[4]} = X_{19}[1]_{[4]}$;

然后更新计数器 $V_2[y_2] += V_1[y_1]$. 此步需要 $2^{56} \times 2^{36} \times 2^4$ 次内存访问.

(c)建立计数器 $V_3[y_3]$,且置为零. 取

$$y_3 = Y_{19}[3,4,7,11,13,15,16] \parallel Y_{19}[12]_{[2,3,4]} \parallel Y_{18}[1]_{[1,4]} \parallel Y_{18}[5]_{[1,4]} \parallel Y_{18}[8]_{[1,3,4]} \parallel I_3[5]_{[4]}$$

穷举4比特密钥 $f_2(k_1)[9]$,计算 $k_0[9]$ 并更新

$I_3[5]_{[4]} = I_3[5]_{[4]} \oplus S(I_3[9] \oplus f_2(k_1)[9])_{[4]}$;

$X_{19}[9] = S(Y_{19}[9] \oplus k_0[9])$;

$Y_{18}[1]_{[1]} = Y_{18}[1]_{[1]} \oplus Y_{19}[9]_{[1]}$;

$Y_{18}[5]_{[1]} = Y_{18}[5]_{[1]} \oplus X_{19}[9]_{[1]}$;

$Y_{18}[5]_{[4]} = Y_{18}[5]_{[4]} \oplus X_{19}[5]_{[4]}$;

$Y_{18}[9]_{[3]} = X_{19}[9]_{[3]}$;

$Y_{18}[9]_{[4]} = Y_{18}[9]_{[4]} \oplus X_{19}[9]_{[4]}$;

**表4 19轮 PRIDE 算法的线性分析**

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $I_1$ | ???? | ???? | 0000 | 0000 | ???? | ???? | 0000 | 0000 | ???? | 0000 | 0000 | ???? | ???? | 0000 | 0000 | ???? |
| $X_1$ | ???? | ???? | 0000 | 0000 | ???? | ???? | 0000 | 0000 | ???? | 0000 | 0000 | ???? | ???? | 0000 | 0000 | ???? |
| $Y_1$ | ???? | 00?? | 0000 | 0000 | ???? | 0?? 0 | 0000 | 0000 | ? 0?? | 0000 | 0000 | 0?? 0 | ?? 0? | 0000 | 0000 | 0?? 0 |
| $Z_1$ | ? 000 | ? 000 | ? 000 | ? 000 | ?? 00 | ?? 00 | 000? | ? 00? | ?? 00 | ?? 00 | ? 00? | 000? | ? 000 | ? 000 | ? 000 | ? 000 |
| $W_1$ | ? 000 | ? 000 | ? 000 | 0000 | ? 000 | ? 000 | ? 000 | 0000 | ? 000 | ? 000 | ? 000 | 0000 | ? 000 | ? 000 | ? 000 | 0000 |
| $I_2$ | ???? | 0000 | 0000 | 0000 | ???? | 0000 | 0000 | 0000 | ???? | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| $X_2$ | ???? | 0000 | 0000 | 0000 | ???? | 0000 | 0000 | 0000 | ???? | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| $Y_2$ | 0001 | 0000 | 0000 | 0000 | 0001 | 0000 | 0000 | 0000 | 0001 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| $Z_2$ | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0010 | 1000 | 1000 | 0000 |
| $W_2$ | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 1000 | 0000 | 0000 |
| $I_3$ | 0000 | 0000 | 0000 | 0000 | 0001 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| $X_{18}$ | 0001 | 0000 | 0000 | 0000 | 0001 | 0000 | 0000 | 0000 | 0001 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| $Y_{18}$ | ???? | 0000 | 0000 | 0000 | ???? | 0000 | 0000 | 0000 | ???? | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| $Z_{18}$ | ? 000 | ? 000 | ? 000 | 0000 | ? 000 | ? 000 | ? 000 | 0000 | ? 000 | ? 000 | ? 000 | 0000 | ? 000 | ? 000 | ? 000 | 0000 |
| $W_{18}$ | ? 000 | ? 000 | ? 000 | ? 000 | 00?? | 00?? | 00?? | ? 0? 0 | 00?? | 00?? | ? 0? 0 | 00?? | ? 000 | ? 000 | ? 000 | ? 000 |
| $I_{19}$ | ? 00? | 0000 | ? 00? | 0?? 0 | ? 00? | 0000 | 0?? 0 | 0?? 0 | ? 0?? | 0000 | 0?? 0 | 0? 00 | ?? 0? | 0000 | 00? 0 | 00? 0 |
| $X_{19}$ | ? 00? | 0000 | ? 00? | 0?? 0 | ? 00? | 0000 | 0?? 0 | 0?? 0 | ? 0?? | 0000 | 0?? 0 | 0? 00 | ?? 0? | 0000 | 00? 0 | 00? 0 |
| $Y_{19}$ | ???? | 0000 | ???? | ???? | ???? | 0000 | ???? | ???? | ???? | 0000 | ???? | 0??? | ???? | 0000 | ???? | ???? |
| $O_{19}$ | ???? | 0000 | ???? | ???? | ???? | 0000 | ???? | ???? | ???? | 0000 | ???? | 0??? | ???? | 0000 | ???? | ???? |

然后更新计数器 $V_3[y_3] += V_2[y_2]$. 此步大体需要 $2^{50}\times 2^{40}\times 2^4$ 次内存访问.

($d$)依次穷举密钥 $k_0[3,4,7,8,11,13,15,16]$, $k_0[12]_{[2,3,4]}$, $\{P^{-1}f_{19}(k_1)\}[1,5,9]$,更新计数器 $V_4[y_4]$,其中

$$y_4 = I_3[5]_{[4]},$$

此步大体需要次内存访问.

**(3)密钥筛选阶段**　若 $y_4=0$,令 $V[k] += V_4[y_4]$. 若 $y_4=1$,令 $V[k] -= V_4[y_4]$. 保留绝对值最高的 $2^{91-a}$ 个密钥为候选密钥.

**复杂度估计**　攻击选择 $N=4\times\varepsilon^{-2}=2^{62}$ 个明密文对. 在分析阶段,第($a$)步需要 $2^{62}\times 2^{36}$ 次内存访问,第($b$)步需要 $2^{62}\times 2^{36}\times 2^4$ 次内存访问;第($c$)步需要 $2^{50}\times 2^{40}\times 2^4$ 次内存访问;第($d$)步需要不超过 $14\times 2^{95}$ 次的内存访问;若访问一次内存等同于 1 轮加密,则一共需要不超过 $2^{95}$ 次 19 轮加密. 分析过程中一共涉及 91 比特密钥,所以存储复杂对为 $2^{88}$ 字节. 由文献[22]可知,成功概率为 87%. 另外,利用第 2 条 1 轮迭代的线性逼近迭代 15 轮作为线性区分器,往前后各扩展 2 轮,对 19 轮 PRIDE 算法做分析,时间复杂度和 18 轮的分析相当,但几乎需要所有的明密文对.

## 7　结束语

本文主要评估了 PRIDE 密码算法关于线性分析方法的安全性. 首先构造出 16 条优势为 $2^{-5}$ 的 2 轮线性逼近和 8 条优势为 $2^{-3}$ 的 1 轮线性逼近. 然后,利用线性分析,结合密钥扩展算法和部分和技术,对 18 轮和 19 轮 PRIDE 密码算法做安全性分析;其中,我们可以利用推导出的 S 盒的线性性质减少分析过程中的计算复杂度. 由结果比对表 5 知,本文的两个结果在计算量没有优于差分分析,但在对 $S$ 盒差分和线性性质联系的探讨、线性层对角对合性质的分析等方面都有理论意义.

**表 5**　PRIDE 算法的主要分析结果

| 分析方法 | 轮数 | 数据复杂度 | 时间复杂度 | 工作出处 |
|---|---|---|---|---|
| 相关密钥差分分析 | 20 | $2^{39}$ CPs | $2^{60}$ ENC | 文献[9] |
| 差分分析 | 18 | $2^{60}$ CPs | $2^{66}$ ENC | 文献[7] |
| 差分分析 | 19 | $2^{62}$ CPs | $2^{63}$ ENC | 文献[8] |
| 线性分析 | 18 | $2^{60}$ KPs | $2^{86}$ ENC | 本文 |
| 线性分析 | 18 | $2^{62}$ KPs | $2^{60}$ ENC | 本文 |
| 线性分析 | 19 | $2^{62}$ KPs | $2^{95}$ ENC | 本文 |
| 线性分析 | 19 | $2^{64}$ KPs | $2^{86}$ ENC | 本文 |

**说明**:CPs 表示选择明文;KPs 表示已知明文;ENC 表示相应轮数的加密.

### 参考文献


[1] Bogdanov A, Knudsen LR, Leander G, et al. Present: An ultra-lightweight block cipher[A]. CHES 2007[C]. Vienna, Austria: LNCS 4727, 2007. 450 – 466.

[2] Izadi M, Sadeghiyan B, Sadeghian S S, et al. MIBS: A new light-weight block cipher[A]. Cryptology and Network Security 2009[C]. LNCS 5888, 2009. 334 – 348.

[3] Guo J, Peyein T, Poschmann A, et al. The LED Block Cipher[A]. CHES 2011[C]. Nara, Japan: LNCS 6917, 2011. 326 – 341.

[4] Wu W L, Zhang L. LBlock: A Lightweight Block Cipher [A]. ACNS 2011[C]. Nerja, Spain: LNCS 6715, 2011. 327 – 344.

[5] Beaulieu R, et al. The SIMON and SPECK Families of Lightweight Block Ciphers[Z]. IACR Cryptology ePrint Archive, 2013: 414.

[6] Albrecht M R, Driessen B, Kavun E B, et al. Block Ciphers-Focus on the Linear Layer (Feat. PRIDE)[M]. Advances in Cryptology-CRYPTO 2014, Santa Barbara, CA, USA: LNCS 8616, 2014. 57 – 76.

[7] Zhao J, et al. Differential Analysis on Block Cipher PRIDE [Z]. Cryptology ePrint Archive, http://eprint. iacr. org/2014/5/25.

[8] Yang Q, et al. Improved differential analysis of block cipher PRIDE[A]. Information Security Practice and Experience[C]. LNCS, 2015. 9065: 209 – 219.

[9] Dai Y, et al. Cryptanalysis of Full PRIDE Block Cipher [Z]. Cryptology ePrint Archive, http://eprint. iacr. org/2014/987. pdf.

[10] Matsui M. Linear cryptanalysis method for DES cipher [A]. Advances in Cryptology-EUROCRYPT' 93[C]. Norway: LNCS 765, 1994. 386 – 397.

[11] Matsui M. The first experimental cryptanalysis of the Data Encryption Standard[A]. Advances in Cryptology-Crypto'94[C]. California, USA: LNCS 839, 1994. 1 – 11.

[12] Biryukov A, De Canniere C, Quisquater M. On multiple linear approximations[A]. Advances in Cryptology-CRYPTO 2004[C]. California, USA: LNCS 3152, 2004. 1 – 22.

[13] Hermelin M, Cho J Y, Nyberg K. Multidimensional linear cryptanalysis of reduced round Serpent[A]. Information Security and Privacy[C]. LNCS 5107, 2008. 203 – 215.

[14] Kaliski Jr B S, Robshaw M J B. Linear cryptanalysis using multiple approximations[A]. Advances in Cryptology-Crypto' 94[C]. California, USA: LNCS 839, 1994. 26 – 39.

[15] Hermelin M, Cho J Y, Nyberg K. Multidimensional extension of Matsui' s algorithm 2[A]. Fast Software Encryption – FSE2009[C]. Leuven, Belgium: LNCS 5665, 2009. 209 – 227.

[16] Bogdanov A, et al. Linear hulls with correlation zero and linear cryptanalysis of block ciphers[J]. Designs, Codes

and Cryptography,2014,70(3):369 – 383.
[17] Bogdanov A,Wang M. et al. Zero correlation linear cryptanalysis with reduced data complexity[A]. Fast Software Encryption – FSE 2012[C]. Washington DC,USA:LNCS 7549,2012. 29 – 48.
[18] Bogdanov A, Leander G, Nyberg K, et al. Integral and multidimensional linear distinguishers with correlation zero[A]. Advances in Cryptology-ASIACRYPT 2012[C]. Beijing:LNCS 7658,2012. 244 – 261.
[19] Biham E, et al. An Improvement of linear cryptanalysis with addition operations with applications to FEAL-8X [A]. Selected Areas in Cryptography – SAC 2014[C]. Montreal,Quebec:LNCS 8781,2014. 59 – 76.
[20] Cho J,et al. Linear cryptanalysis of reduced-round PRESENT[A]. Topics in Cryptology-CT-RSA 2010[C]. San Francisco,Ca:LNCS 5985,2010. 302 – 317.
[21] 陈怀凤,温隆,王美琴. 基于 FFT 技术的 MULTI2 线性分析[J]. 密码学报,2014,1(4):311 – 320.
Chen Huaifeng,Wen Long,Wang Meiqin. Linear cryptanalysis on MULTI2 with FFT technique[J]. Journal of Cryptologic Research,2014,1(4):311 – 320. (in Chinese)
[22] Selcuk A A,et al. On probability of success in linear and differential cryptanalysis[J]. Journal of Cryptology,2008, 21(1):131 – 147.

## 作者简介

**伊文坛** 男,1989 年生于山东菏泽. 研究方向为分组密钥安全性分析.
E-mail:nlwt89@ sina. com

**田 亚** 男,1991 年生于江苏徐州. 硕士研究生,研究方向为分组密码安全性分析.

**陈少真** 女,1967 年生于江苏无锡. 博士生导师,研究方向为分组密码的设计与分析.