计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.200200034

PFP 算法改进的不可能差分分析

沈 璇 王欣玫 何 俊 孙志远
国防科技大学信息通信学院 武汉 430010
(shenxuan_08@163.com)

摘 要 目前资源受限环境的应用场景越来越多,该场景下的数据加密需求也随之增加。以国际标准 PRESENT 算法为代表的一大批轻量级分组密码应运而生。PFP 算法是一种基于 Feistel 结构的超轻量级分组密码算法,它的轮函数设计借鉴了国际标准 PRESENT 算法的设计思想。PFP 算法的分组长度为 64 比特,密钥长度为 80 比特,迭代轮数为 34 轮。针对 PFP 算法,研究了其抵抗不可能差分分析的能力。在该算法的设计文档中,设计者利用 5 轮不可能差分区分器攻击 6 轮的 PFP 算法,能够恢复 32 比特的种子密钥。与该结果相比,文中通过研究轮函数的具体设计细节,利用 S 盒的差分性质构造出 7 轮不可能差分区分器,并攻击 9 轮的 PFP 算法,能够恢复 36 比特的种子密钥。该结果无论在攻击轮数还是恢复的密钥量方面,均优于已有结果,是目前 PFP 算法最好的不可能差分分析结果。

关键词:分组密码;PRESENT 算法;PFP 算法;不可能差分分析;非线性组件
中图法分类号 TP309;TN918

Revised Impossible Differential Cryptanalysis of PFP Block Cipher

SHEN Xuan, WANG Xin-mei, HE Jun and SUN Zhi-yuan
College of Information and Communication, National University of Defense Technology, Wuhan 430010, China

Abstract Nowadays, the application scenarios in the resource-constrained terminal system appear more and more, and the data encryption requirement of them also needs to be satisfied. There are many lightweight block ciphers designed such as PRESENT which is an international standard block cipher. PFP cipher is an ultra-lightweight block cipher which takes Feistel structure, and its round function is designed by using the experience of PRESENT cipher for reference. The block size of PFP is 64-bit, the key size of PFP is 80-bit and its round number is 34. For PFP, this paper studies its ability against impossible differential cryptanalysis. In the design document, the designers proposed a 5-round impossible differential and attacked reduced 6-round PFP cipher with this distinguisher. Moreover, the designers can recover 32-bit master key. Comparing with this result, by exploiting the differential property of the S-box in PFP, this paper constructs a 7-round impossible differential distinguisher and attack reduced 9-round PFP. Moreover, it can recover 36-bit master key. Therefore, the result is much better than the known one in terms of either the round number or the recovered key. So far as I know, the result in this paper is the best impossible differential cryptanalysis of PFP cipher.

Keywords Block cipher, PRESENT algorithm, PFP algorithm, Impossible differential cryptanalysis, Non-linear component

1 前言

随着计算机技术的不断发展,资源受限环境下的数据加密需求越来越强烈。传统的分组密码算法,如高级加密标准 AES 算法[1]等,受限于资源条件,并不能很好地直接应用于这些资源受限的环境中。为了解决该问题,近年来许多密码学者相继提出了适用于资源受限环境的轻量级分组密码算法,如 HIGHT, PRESENT, LED, LBlock, SIMECK, SKINNY, GIFT 等[2-8]。受国际标准 PRESENT 算法的启发,Huang 等提出了一种新的轻量级分组密码算法——PFP 算法[9]。该算法整体上采用 Feistel 结构,轮函数设计采用类似于 PRESENT 算法的 SP 结构。

分组密码算法的分析方法主要有两大类:差分分析和线性分析。其中,差分分析主要是利用高概率的差分来恢复密钥。根据差分分析的思想,密码学学者们陆续提出了很多密码分析方法,如截断差分分析、多重差分分析、不可能差分分析等[10-12]。不可能差分分析是目前针对分组密码算法最有效的分析方法之一,它对 AES[13], ARIA[14], Midori[15], GRANULE[16]等分组密码算法的分析效果显著。该分析方法由 Knudsen[17]和 Biham 等[12]独立提出。不可能差分分析

到稿日期:2020-02-05 返修日期:2020-05-30 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:国家自然科学基金(61902414)
This work was supported by the National Natural Science Foundation of China(61902414).
通信作者:何俊(3439992@qq.com)

的思想与差分分析相反，它是利用概率为零的差分来筛除错误密钥，进而得到正确密钥。该分析方法由不可能差分区分器的构造和密钥恢复两部分构成。首先要构造尽可能长的不可能差分区分器，再利用已构造的区分器进行密钥恢复。

本文主要研究了 PFP 算法抵抗不可能差分分析的能力。在 PFP 算法的设计文档中，设计者给出了该算法的不可能差分分析结果，并利用构造的 5 轮不可能差分区分器攻击了 6 轮 PFP 算法，能够恢复 32 比特的种子密钥。本文通过研究 PFP 算法轮函数中 S 盒的差分性质，构造出 7 轮不可能差分区分器，并在区分器首尾各加一轮，攻击了 9 轮的 PFP 算法。本文的攻击结果能够恢复 36 比特的种子密钥，数据复杂度为 2^{36} 个选择明文，时间复杂度为 2^{58} 次 9 轮算法加密。与已有结果相比，本文的结果不论在攻击轮数、复杂度，还是恢复的密钥量方面，均优于文献[9]给出的结果。

本文第 2 节介绍 PFP 算法的整体框架和轮函数的加密细节；第 3 节利用 S 盒的差分性质，构造 PFP 算法 7 轮的不可能差分区分器；第 4 节利用构造的区分器攻击 9 轮的 PFP 算法；第 5 节将本文结果与文献[9]给出的结果进行对比分析；最后总结全文，并对后续工作进行展望。

2 PFP 算法

2.1 PFP 算法的整体框架

PFP 算法整体采用 Feistel 结构，轮函数采用类似于 PRESENT 算法的 SP 结构，其中 S 盒为 4 比特，P 置换采用 32 比特的比特拉线设计。PFP 算法的分组长度为 64 比特，密钥长度为 80 比特，算法总轮数为 34 轮。该算法的整体加密流程如图 1 所示。

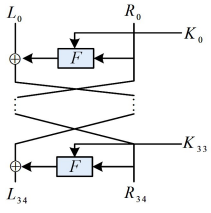

图 1 PFP 算法的加密流程

Fig. 1 Encryption process of PFP

在加密时，将 64 比特的分组数据分为左右各 32 比特，即左支 L_i 和右支 R_i 均为 32 比特。明文 $P=(L_0,R_0)$ 经过 34 轮迭代后得到密文 $C=(L_{34},R_{34})$，最后一轮左右两支不交换。若一轮迭代的输入为 (L_i,R_i)，输出为 (L_{i+1},R_{i+1})，则有：

$$L_{i+1}=R_i;R_{i+1}=L_i\oplus F(R_i,K_i) \tag{1}$$

其中，K_i 表示第 i 轮的轮密钥，F 表示轮函数。

2.2 PFP 算法轮函数

PFP 算法轮函数的设计采用 SP 结构，与国际标准 PRESENT 算法类似。轮函数 F 的加密流程由轮密钥加、S 层和 P 置换 3 部分组成，如图 2 所示。

轮密钥加：分组加密数据的右支 R_i 与第 i 轮的轮密钥 K_i 进行逐比特异或。

S 层：8 个相同的 4 比特 S 盒并置加密。若 S 层的输入为 $(a_7|a_6|a_5|a_4|a_3|a_2|a_1|a_0)$，输出为 $(b_7|b_6|b_5|b_4|b_3|b_2|b_1|b_0)$，则有 $b_i=S(a_i)$，其中 a_i 和 $b_i(i=0,1,\cdots,7)$ 均为 4 比特，“|”表示比特串之间的连接。PFP 算法中的 4 比特 S 盒与国际标准 PRESENT 算法的 S 盒相同，如表 1 所列。表 1 中数字均为 16 进制表示，例如 $S(0x0)=0xC$。

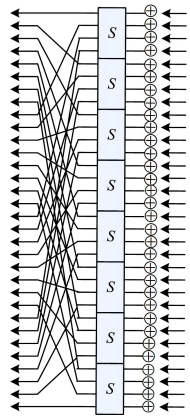

图 2 轮函数 F 的加密流程

Fig. 2 Encryption process of round function F

表 1 PFP 算法的 S 盒

Table 1 S-box of PFP

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

P 置换：按照比特拉线设计，即 32 比特数据按照表 2 的规律进行比特置换。在表 2 中，输入的第 i 比特经过 P 置换后变为第 $P(i)$ 比特。

表 2 PFP 算法的 P 置换

Table 2 Permutation of PFP

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
$P(i)$	0	8	16	24	1	9	17	25	2	10	18	26	3	11	19	27	4	12	20	28	5	13	21	29	6	14	22	30	7	15	23	31

在研究不可能差分区分器的构造时，由于轮密钥不影响差分的传播，因此本文不详细介绍 PFP 的密钥扩展算法，具体细节见文献[9]。

3 PFP 算法 7 轮不可能差分区分器的构造

构造尽可能长的不可能差分区分器是不可能差分分析的核心。在 PFP 算法中，设计者利用中间相错技术构造了 PFP 算法 5 轮不可能差分区分器。在该区分器的构造中，设计者仅利用了轮函数为双射的性质，并没有充分利用轮函数的具体细节。

本节首先挖掘 PFP 算法 S 盒的具体细节，给出一个关于 S 盒的差分性质；然后利用该性质，结合 P 置换的加密流程，构造出 PFP 算法的一条 7 轮不可能差分区分器。该区分器的长度比文献[9]构造的长 2 轮。

根据表 1 给出的 PFP 算法的 S 盒，利用计算机程序搜索容易得出该 S 盒的差分分布表，如表 3 所列。

表 3　PFP 算法 S 盒的差分分布表

Table 3　Differential distribution table of S-box in PFP

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	4	0	0	0	4	0	4	0	0	0	4	0	0
2	0	0	0	2	0	4	2	0	0	0	2	0	2	2	2	0
3	0	2	0	2	2	0	4	2	0	0	2	2	0	0	0	0
4	0	0	0	0	0	4	2	2	0	2	2	0	2	0	2	0
5	0	2	0	0	2	0	0	0	0	2	2	2	4	2	0	0
6	0	0	2	0	0	0	2	0	2	0	0	4	2	0	0	4
7	0	4	2	0	0	0	2	0	2	0	0	0	2	0	0	4
8	0	0	0	2	0	0	0	2	0	2	0	4	0	2	0	4
9	0	0	2	0	4	0	2	0	2	0	0	0	2	0	4	0
A	0	0	2	2	0	4	0	0	2	0	2	0	0	2	2	0
B	0	2	0	0	2	0	0	0	4	2	2	2	0	2	0	0
C	0	0	2	0	0	4	0	2	2	2	2	0	0	0	2	0
D	0	2	4	2	2	0	0	2	0	0	2	2	0	0	0	0
E	0	0	2	2	0	0	2	2	2	2	0	0	2	2	0	0
F	0	4	0	0	4	0	0	0	0	0	0	0	0	0	4	4

注：$ 表示数字 16

在表 3 中，第 1 列为 S 盒的输入差分，第 1 行为 S 盒的输出差分，输入与输出差分均用 16 进制表示，中间的值为该输入输出差分方程对应的解的个数。例如，当 S 盒的输入差分为 0x1 时，输出差分为 0xD 时，满足该差分方程的解的个数为 4。注意当解的个数为 0 时，说明该输入差分和输出差分不匹配，即该输入差分经过 S 盒后不能传播到该输出差分。

根据 PFP 算法 S 盒的差分分布表，可以得到如下性质。

性质 1　对于 PFP 算法的 S 盒，当输入差分分别为 0x1，0x8，0x9 时，它们经过 S 盒后输出差分的最低比特是确定的，即有如下 3 条差分传播以概率 1 成立：

0x1→ * * * 1

0x8→ * * * 1

0x9→ * * * 0

其中，* 表示该比特差分不确定，可能为 0 或 1。

利用 PFP 算法 S 盒的差分分布表容易验证上述性质，下面以 0x1→ * * * 1 为例进行分析。根据表 3，当输入差分为 0x1 时，输出差分可能为 0x3，0x7，0x9，0xD，则输出差分前 3 比特 0 或 1 均能够出现，但是最低比特差分一定为 1，即输出差分的形式为 * * * 1。因此，0x1→ * * * 1 以概率 1 成立。类似地，可以说明 0x8→ * * * 1 和 0x9→ * * * 0 以概率 1 成立。

下面利用性质 1 并结合 P 置换的特点，给出定理 1。

定理 1　在 PFP 算法中，当输入差分为$(\alpha_0|0_{32})$，输出差分为$(0_{32}|\beta_0)$，并且 α_0 满足$(*_{16}|0_{12}|100*)$，β_0 满足$(*_{16}|0_{16})$的形式时，$(\alpha_0|0_{32})\rightarrow(0_{32}|\beta_0)$是 PFP 算法的一条 7 轮不可能差分区分器。其中，* 表示该位置的比特差分不确定，$*_i$ 表示连续 i 个 *，0_i 表示连续 i 个 0，α_0 和 β_0 均为 32 比特的差分。

证明：先通过差分方程求解，给出$(\alpha_0|0_{32})\rightarrow(0_{32}|\beta_0)$是 PFP 算法 7 轮可能差分的必要条件，再利用轮函数的具体细节，并结合性质 1，证明当 α_0 满足$(*_{16}|0_{12}|100*)$，β_0 满足$(*_{16}|0_{16})$的形式时，该必要条件不成立，使得定理得证。

当输入差分为$(\alpha_0|0_{32})$时，它从加密方向经过 4 轮差分传播的规律如表 4 所列。在表 4 中，$\alpha_i(i=1,2,3)$均为 32 比特差分，$\Delta F^r(\alpha_i)$表示当输入差分为 α_i 时，经过连续 r 轮 F 函数后所有可能输出差分的集合。

表 4　加密方向的 4 轮差分传播

Table 4　4-round differential propagation from encryption direction

轮数	左支	右支	备注
0	α_0	0_{32}	
1	0_{32}	α_0	
2	α_0	α_1	$\alpha_1\in\Delta F(\alpha_0)$
3	α_1	α_2	$\alpha_2\in\Delta F^2(\alpha_0)\oplus\alpha_0$
4	α_2	α_3	$\alpha_3\in\Delta F(\alpha_2)\oplus\alpha_1$

类似地，当输出差分为$(0_{32}|\beta_0)$时，它从解密方向经过 3 轮差分传播的规律如表 5 所列。其中，$\beta_i(i=1,2)$均为 32 比特差分。

表 5　解密方向的 3 轮差分传播

Table 5　3-round differential propagation from decryption direction

轮数	左支	右支	备注
0	0_{32}	β_0	
1	β_0	0_{32}	
2	β_1	β_0	$\beta_1\in\Delta F(\beta_0)$
3	β_2	β_1	$\beta_2\in\Delta F^2(\beta_0)\oplus\beta_0$

若$(\alpha_0|0_{32})\rightarrow(0_{32}|\beta_0)$是 PFP 算法一条 7 轮可能的差分，则它需要满足以下两个差分方程：

$$\begin{cases}\alpha_2=\beta_2\\ \alpha_3=\beta_1\end{cases}\tag{2}$$

因此，当$(\alpha_0|0_{32})\rightarrow(0_{32}|\beta_0)$是 PFP 算法 7 轮可能差分时，则一定有 $\alpha_2=\beta_2$ 成立。该命题的逆否命题即为：若 $\alpha_2\neq\beta_2$，则$(\alpha_0|0_{32})\rightarrow(0_{32}|\beta_0)$一定是 PFP 算法的一条 7 轮不可能差分。

下面说明当 α_0 满足$(*_{16}|0_{12}|100*)$，β_0 满足$(*_{16}|0_{16})$的差分形式时，$\alpha_2\neq\beta_2$。

当输入差分 α_0 满足$(*_{16}|0_{12}|100*)$的形式时，根据最低比特差分为 0 或 1，分以下两种情况进行讨论。

(1)当 α_0 满足$(*_{16}|0_{12}|1000)$的形式时，利用性质 1，结合轮函数的差分传播可知，$\Delta F^2(\alpha_0)$一定为$(*_{31}|1)$，如图 3 所示。

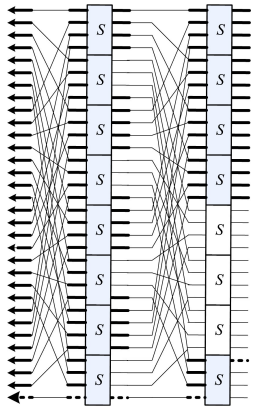

图 3　输入差分为$(*_{16}|0_{12}|1000)$时，两轮轮函数的差分传播

Fig. 3　2-round differential propagation with input difference $(*_{16}|0_{12}|1000)$

其中，粗黑线段表示该比特差分为 *，细黑线段表示该比特差分为 0，虚线表示该比特差分为 1。因此，$\alpha_2=\Delta F^2(\alpha_0)\oplus\alpha_0$ 的最低比特一定为 1。

(2)当 α_0 满足$(*_{16}|0_{12}|1001)$的形式时，利用性质 1，结合轮函数的差分传播可知，$\Delta F^2(\alpha_0)$一定为$(*_{7}0|*_{7}0|*_{7}0|*_{7}0)$，如图 4 所示。

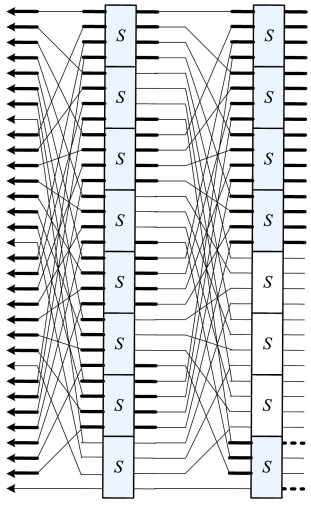

图 4 输入差分为($*_{16}|0_{12}|1001$)时,两轮轮函数的差分传播

Fig. 4 2-round differential propagation with input difference ($*_{16}|0_{12}|1001$)

其中不同线段表示的含义与图 3 相同。因此 $\alpha_2 \in \Delta F^2(\alpha_0) \oplus \alpha_0$ 的最低比特一定为 1。

因此,当输入差分 α_0 满足($*_{16}|0_{12}|100*$)时,α_2 的最低比特差分一定为 1。

当输出差分 β_0 为($*_{16}|0_{16}$)时,连续经过两轮轮函数后的输出差分形式 $\Delta F^2(\beta_0)$ 为($*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0$),如图 5 所示,其中不同线段表示的含义与图 3 相同。因此,$\beta_2 = \Delta F^2(\beta_0) \oplus \beta_0$ 的最低比特一定为 0。

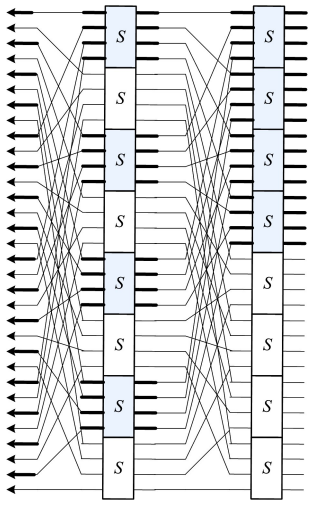

图 5 输入差分为($*_{16}|0_{16}$)时,两轮轮函数的差分传播

Fig. 5 2-round differential propagation with intput difference ($*_{16}|0_{16}$)

综上所述,当输入差分 α_0 满足($*_{16}|0_{12}|100*$),输出差分 β_0 满足($*_{16}|0_{16}$)的形式时,α_2 的最低比特差分一定为 1,β_2 的最低比特差分一定为 0,故 $\alpha_2 \neq \beta_2$。此时,$(\alpha_0|0_{32}) \nrightarrow (0_{32}|\beta_0)$ 是 PFP 算法的一条 7 轮不可能差分区分器。定理得证。

4 PFP 算法 9 轮不可能差分分析

本节在上节构造的 7 轮不可能差分区分器的基础上前后各加一轮,攻击 9 轮 PFP 算法。

图 6 为攻击 9 轮 PFP 算法的示意图。

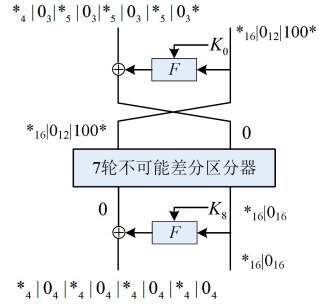

图 6 PFP 算法 9 轮不可能差分攻击

Fig. 6 Impossible differential attack on reduced 9-round PFP

步骤 1 选取一个结构,在该结构中任意选择两个明文进行异或,它们均满足如下差分:

($*_4|0_3|*_5|0_3|*_5|0_3|*_5|0_3*, *_{16}|0_{12}|100*$)

该结构能够提供 $2^{(4+5+5+5+1+16+1)} = 2^{37}$ 个明文和 $2^{37} \times 2^{37}/2 = 2^{73}$ 组明文对。

步骤 2 选择 2^n 个上述结构,则这些结构总共能够提供 2^{n+37} 个明文和 2^{n+73} 组明文对。选择使得密文差分满足以下形式的密文对:

($*_4|0_4|*_4|0_4|*_4|0_4|*_4|0_4, *_{16}|0_{16}$)

在上述密文差分中共 32 比特差分为 0,故满足上述密文差分形式的概率为 2^{-32}。因此,经过步骤 2 的筛选后,剩下的明密对数目约为 $2^{n+73} \times 2^{-32} = 2^{n+41}$。

步骤 3 对于经过上述步骤处理的剩余明密对,猜测最后一轮的轮密钥 K_8 中的高 16 比特密钥,即 $K_{8,31}, K_{8,30}, \cdots, K_{8,16}$,使得关于高 16 比特的 4 个 S 盒差分方程成立。在平均意义下,每个 4 比特 S 盒差分方程成立的概率约为 2^{-4},则 4 个差分方程同时成立的概率为 $2^{-4\times4}$。因此,经过该步骤后,剩余明密对的数目约为 $2^{n+41} \times 2^{-16} = 2^{n+25}$。

步骤 4 对于经过上述步骤处理的剩余明密对,猜测第一轮轮密钥 K_0 的高 16 比特以及最低 4 比特,即 $K_{0,31}, K_{0,30}, \cdots, K_{0,16}, K_{0,3}, K_{0,2}, K_{0,1}, K_{0,0}$。这 20 比特的密钥共涉及 5 个 4 比特 S 盒差分方程,而每个 S 盒差分方程的概率约为 2^{-4},则 5 个差分方程同时成立的概率为 $2^{-4\times5} = 2^{-20}$。若它们同时成立,则筛除步骤 3 和步骤 4 中猜测的 16+20=36 比特密钥值。

注意到中间的 7 轮差分是不可能差分,当猜测的 36 比特候选密钥满足不可能差分的输入输出时,猜测的密钥一定是错误的。在分析了约 2^{n+25} 组明密对后,仍然能够保留下来的错误密钥数目为:

$$(2^{36}-1) \times (1-2^{-20})^{2^{n+25}} \approx 2^{36} \times e^{-2^{n+5}} \approx 2^{36-1.44\times2^{n+5}} \tag{3}$$

当 $n=0$ 时:

$$2^{36-1.44\times2^{n+5}} = 2^{36-46.08} = 2^{-14.08} < 1 \tag{4}$$

此时,错误密钥被淘汰,剩下的即为正确密钥。

通过上述方法可以恢复 K_0 的 20 比特和 K_8 的 16 比特(共计 36 比特)密钥信息,再根据 PFP 的密钥扩展算法可以得到种子密钥的 36 比特信息。

复杂度分析:在上述攻击中,当 $n=0$ 时,数据复杂度为 $2^{n+36} = 2^{36}$ 个选择明文。时间复杂度主要由步骤 3 和步骤 4 提

供。在步骤 3 中，解密过程涉及 4 个 S 盒，而 1 轮解密共涉及 8 个 S 盒，因此步骤 3 的时间复杂度为 $2\times 2^{16}\times 2^{n+41}\times 4/8=2^{n+57}=2^{57}$ 次 1 轮解密；在步骤 4 中，加密过程涉及 5 个 S 盒，而 1 轮加密共涉及 8 个 S 盒，因此步骤 4 的时间复杂度为 $2\times 2^{36}\times 2^{n+25}\times(5/8)\approx 2^{n+61}=2^{61}$ 次 1 轮加密。故总的时间复杂度为 $2^{57}+2^{61}\approx 2^{61}$ 次 1 轮加密，即 $2^{61}\times 1/9\approx 2^{58}$ 次 9 轮 PFP 算法加密。

5　结果对比

本文充分利用了轮函数的具体细节，特别是挖掘非线性组件 S 盒的差分性质，并结合 P 置换设计，利用差分方程求解的方法构造出 PFP 算法的 7 轮不可能差分区分器，比文献[9]给出的 5 轮区分器长 2 轮。进一步，本文利用构造的区分器攻击了 9 轮 PFP 算法，攻击的数据复杂度为 2^{36} 个选择明文，时间复杂度为 2^{58} 次 9 轮算法加密。此外，该攻击能够恢复 36 比特的种子密钥信息。本文攻击结果与已有攻击结果的比较如表 6 所列。可以看出，本文构造的区分器轮数、攻击轮数以及恢复的种子密钥量均优于已有文献的结果；此外，攻击所需的复杂度也更低。

表 6　PFP 算法不可能差分攻击的比较

Table 6　Comparison of impossible differential attack on PFP

不可能差分攻击	区分器轮数	攻击轮数	恢复密钥量	数据复杂度	时间复杂度
文献[9]	5 轮	6 轮	32 比特	2^{37}	2^{65}
本文	7 轮	9 轮	36 比特	2^{36}	2^{58}

结束语　本文研究了 PFP 算法的不可能差分性质。通过挖掘轮函数的信息，本文构造了 PFP 算法 7 轮不可能差分区分器，并利用该区分器攻击了 9 轮 PFP 算法。相比文献[9]给出的不可能差分结果，本文的攻击结果无论在攻击轮数、复杂度还是恢复的密钥量方面均更优。目前，本文结果是关于 PFP 算法不可能差分分析最好的结果。此外，PFP 算法采用的非线性组件 S 盒与国际标准 PRESENT 算法相同，后续我们将继续探索利用本文提出的 S 盒差分性质对国际标准 PRESENT 算法进行安全性分析。

参 考 文 献

[1] DAEMEN J，RIJMEN V. The Design of Rijndael：AES-the Advanced Encryption Standard[M]. Berlin：Springer-Verlag，2002：31-148.

[2] HONG D，SUNG J，HONG S，et al. HIGHT：a new block cipher suitable for low-resource device[C]// Proceedings of the 2006 International Workshop on Cryptographic Hardware and Embedded Systems. Yokohama，Japan，2006：46-59.

[3] BOGDANOV A，KNUDSEN L R，LEANDER G，et al. PRESENT：an ultra-lightweight block cipher[C]// Proceedings of the 2007 International Workshop on Cryptographic Hardware and Embedded Systems. Vienna，Austria，2007：450-466.

[4] GUO J，PEYRIN T，POSCHMANN A，et al. The LED block cipher[C]// Proceeding of the 2011 International Workshop on Cryptographic Hardware and Embedded Systems. Nara，Japan，2011：326-341.

[5] WU W L，ZHANG L. LBlock：a lightweight block cipher[C]// Proceedings of the 9th International Conference on Applied Cryptography and Network Security. Nerja，Spain，2011：327-344.

[6] YANG G Q，ZHU B，SUDER V，et al. The Simeck family of lightweight block ciphers[C]// Proceeding of the 2015 International Workshop on Cryptographic Hardware and Embedded Systems. Saint-Malo，France，2015：307-329.

[7] BEIERLE C，JEAN J，KÖLBL S，et al. The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS[C]// Proceeding of the 36th Advances in Cryptology-CRYPTO 2016. Santa Barbara，CA，USA，2016：123-153.

[8] BANIK S，PANDEY S K，PEYRIN T，et al. GIFT：A Small Present[C]// Proceeding of the 2017 International Workshop on Cryptographic Hardware and Embedded Systems. Taipei，Taiwan，2017：321-345.

[9] HUANG Y H，DAI X J，SHI Y Y，et al. Ultra-light weight block cipher algorithm (PFP) based on Feistel structure[J]. Computer Science，2017，44(3)：163-168.

[10] KNUDSEN L R. Truncated and Higher Order Differentials[C]// Proceeding of the Fast Software Encryption-FSE 1994. Leuven：Springer-Verlag，1995：196-211.

[11] BLONDEAU C，GERARD B. Multiple Differential Cryptanalysis：Theory and Practice [C]// Proceeding of the Fast Software Encryption-FSE 2011. Lyngby：Springer-Verlag，2011：35-54.

[12] BIHAM E，BIRYUKOV A，SHAMIR A. Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials [C]// Proceeding of the Advances in Cryptology-EUROCRYT 1999. Prague：Springer-Verlag，1999：12-23.

[13] BOURA C，LALLEMAND V，PLASENCIA M N，et al. Making the impossible possible[J]. Journal of Cryptology，2018，31(1)：101-133.

[14] SHEN X，HE J. Improved Impossible Differential Attack on 7-round Reduced ARIA-256[J]. KSII Transactions on Internet and Information Systems，2019，13(11)：5773-5784.

[15] SASAKI Y，TODO Y. New Impossible Differential Search Tool from Design and Cryptanalysis Aspects[C]// Advances in Cryptology-EUROCRYT 2017. Paris，2017：185-215.

[16] WU X N，LI Y X，WEI Y Z，et al. Impossible differential distinguisher analysis of GRANULE and MANTRA algorithm[J]. Journal on Communications，2020，41(1)：94-101.

[17] KNUDSEN L. DEAL-A 128-bit Block Cipher [R]. University of Bergen，Norway，1998.

SHEN Xuan，born in 1990，Ph.D，lecturer. His main research interests include design and cryptanalysis of symmetric ciphers.

HE Jun，born in 1979，Ph.D，professor. His main research interests include cryptography and network security.