Journal of Computer Applications
计算机应用, 2020, 40(10): 2967 - 2972
ISSN 1001-9081
CODEN JYIIDU
2020-10-10
http://www.joca.cn

文章编号:1001-9081(2020)10-2967-06
DOI:10.11772/j.issn.1001-9081.2019122228

对PICO算法基于可分性的积分攻击

刘宗甫[1,2],袁　征[1*],赵晨曦[1,2],朱　亮[1,2]
(1. 北京电子科技学院 密码科学与技术系,北京 100070; 2. 西安电子科技大学 通信工程学院,西安 710071)
(*通信作者电子邮箱 zyuan@tsinghua.edu.cn)

摘　要:对近年来提出的基于比特的超轻量级分组密码算法PICO抵抗积分密码分析的安全性进行评估。首先,研究了PICO密码算法的结构,并结合可分性质的思想构造其混合整数线性规划(MILP)模型;然后,根据设置的约束条件生成用于描述可分性质传播规则的线性不等式,并借助数学软件求解MILP问题,从目标函数值判断构建积分区分器成功与否;最终,实现对PICO算法积分区分器的自动化搜索。实验结果表明,搜索到了PICO算法目前为止最长的10轮积分区分器,但由于可利用的明文数太少,不利于密钥恢复。为了取得更好的攻击效果,选择搜索到的9轮积分区分器对PICO算法进行11轮密钥恢复攻击。通过该攻击能够恢复128比特轮子密钥,攻击的数据复杂度为$2^{63.46}$,时间复杂度为2^{76}次11轮算法加密,存储复杂度为2^{20}。

关键词:超轻量级分组密码算法;PICO;积分密码分析;可分性质;混合整数线性规划
中图分类号:TP309.7　　**文献标志码**:A

Integral attack on PICO algorithm based on division property

LIU Zongfu[1,2], YUAN Zheng[1*], ZHAO Chenxi[1,2], ZHU Liang[1,2]
(1. *Department of Cryptography Science and Technology, Beijing Electronic Science and Technology Institute, Beijing* 100070, *China*;
2. *School of Communication Engineering, Xidian University, Xi'an Shaanxi* 710071, *China*)

Abstract: PICO proposed in recent years is a bit-based ultra lightweight block cipher algorithm. The security of this algorithm to resist integral cryptanalysis was evaluated. Firstly, by analyzing the structure of PICO cipher algorithm, a Mixed-Integer Linear Programming (MILP) model of the algorithm was established based on division property. Then, according to the set constraints, the linear inequalities were generated to describe the propagation rules of division property, and the MILP problem was solved with the help of the mathematical software, the success of constructing the integral distinguisher was judged based on the objective function value. Finally, the automatic search of integral distinguisher of PICO algorithm was realized. Experimental results showed that, the 10-round integral distinguisher of PICO algorithm was searched, which is the longest one so far. However, the small number of plaintexts available is not conducive to key recovery. In order to obtain better attack performance, the searched 9-round distinguisher was used to perform 11-round key recovery attack on PICO algorithm. It is shown that the proposed attack can recover 128-bit round key, the data complexity of the attack is $2^{63.46}$, the time complexity is 2^{76} 11-round encryptions, and the storage complexity is 2^{20}.

Key words: ultra lightweight block cipher algorithm; PICO; integral cryptanalysis; division property; Mixed-Integer Linear Programming (MILP)

0　引言

轻量级分组密码以实现效率高、加密速度快、软硬件占用资源少而成为最近几年密码领域的研究热点,在计算能力有限,资源受限的环境下被广泛地应用,其中近些年的轻量级分组密码有GIFT[1]、Midori[2]、LED[3]、SKINNY[4]等。因此,对分组密码的安全性分析成为了密码学研究的重要方向之一。

积分分析在2002年FSE(Fast Software Encryption)会议上被提出,作为与差分分析相对应的一种分析方法,它对某些结构和算法的分析有时比差分和线性分析更有效。积分分析结合了Square攻击[5]、Multiset攻击[6]和Saturation攻击[7]的思想,主要是考虑特定输入对应密文某些位置的零和性质,早期主要局限于以字节为单位的密码算法。2008年的FSE会议上,Z'aba等[8]提出了基于比特模式的积分攻击。2015年欧洲密码会议上,Todo[9]提出了可分性的概念,来描述介于"活跃"和"零和"之间的隐含性质。在之后的美洲密码会议上,Todo提出将可分性与S盒的代数标准型进行结合,可分性将发挥更大的优势。基于这一猜想,Todo首次给出了MISTY1算法[10]的全轮破解。比特级可分性质作为可分性的一种特殊情况,可以对算法结构在比特级进行更加精准的刻画。可分性质刚提出来时,由于复杂度的限制,无法在大分组、长轮数算法中应用。为了解决这一问题,2016年亚洲密码会议上,Xing

收稿日期:2020-01-06;**修回日期**:2020-03-25;**录用日期**:2020-03-27。
基金项目:"十三五"国家密码发展基金密码理论课题(MMJJ20180217)。
作者简介:刘宗甫(1995—),男,陕西西安人,硕士研究生,主要研究方向:对称密码算法的安全性分析; 袁征(1968—),女,山西中阳人,教授,博士,主要研究方向:密码设计、密码分析、密码混淆; 赵晨曦(1996—),女,陕西西安人,硕士研究生,主要研究方向:对称密码算法的安全性分析; 朱亮(1995—),男,山东泰安人,硕士研究生,主要研究方向:对称密码算法的安全性分析。

等[11]把追踪可分性的问题转化为基于混合整数线性规划的问题，调用已有的求解器进行基于比特级可分性的自动化搜索，基于混合整数线性规划（Mixed-Integer Linear Programming，MILP）的方法在一定程度上解决复杂度的问题。随后，该方法在积分分析方面得到了广泛应用[12-14]，并取得了显著成效。

对于PICO算法，马楚焱等[15-16]利用MILP方法构造了7轮多维零相关线性区分器，并对10轮PICO算法进行密钥恢复攻击，但目前为止尚没有该算法在积分分析下的安全性评估结果。本文采用MILP搜索工具对PICO算法进行积分区分器的搜索，搜索得到了轮数最长的10轮积分区分器，但由于输入活跃比特数太多，可利用的明文数太少，不利于密钥恢复。本文选择搜索到的9轮积分区分器向后攻击两轮，对PICO算法进行11轮的积分攻击。根据现有研究，这是目前为止首次评估PICO算法在积分攻击方面的安全性。同时，近年来对于密码分析方法之间联系的研究也得到了越来越多的关注，Bogdanov等[17]研究了零相关线性分析和积分分析之间的联系，并从理论上证明了零相关线性区分器可以和积分区分器相互转化。

1 PICO算法

1.1 PICO算法加密过程

PICO算法是2016年Bansod等[18]设计的混淆扩散网络（Substitution Permutation Network，SPN）结构的超轻量级分组密码，它采用64比特明文，128比特主密钥，共迭代32轮，在每一轮轮函数中将64比特明文和64比特轮密钥逐位异或，然后进行列替换和比特置换操作，具体结构如图1所示。算法的64比特明文被排列成4行16列的二维数组形式，具体形式可以用矩阵$\boldsymbol{P}$来表示，其中第i行第j列的元素记为$p^{i,j}$。

$$\boldsymbol{P}=\begin{bmatrix} p^{0,15} & p^{0,14} & \cdots & p^{0,1} & p^{0,0} \\ p^{1,15} & p^{1,14} & \cdots & p^{1,1} & p^{1,0} \\ p^{2,15} & p^{2,14} & \cdots & p^{2,1} & p^{2,0} \\ p^{3,15} & p^{3,14} & \cdots & p^{3,1} & p^{3,0} \end{bmatrix}$$

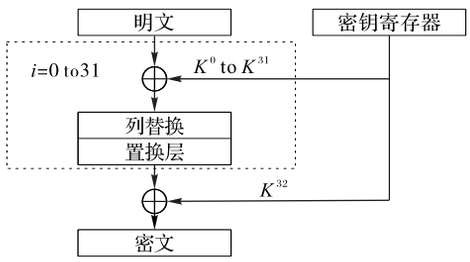

图1 PICO算法结构

Fig. 1 Structure of PICO

PICO算法的每一轮包含以下3个步骤：密钥加、列替换和置换层，在最后一轮仅有轮密钥加。

1）密钥加：64比特中间状态矩阵$\boldsymbol{P}$和64比特轮密钥的对应位置进行异或操作。

$\boldsymbol{P} \rightarrow \boldsymbol{P} \oplus K^i$

2）列替换：对状态矩阵的每一列进行S盒操作，当一个S盒的列输入为$C(i)=p^{3,i}\|p^{2,i}\|p^{1,i}\|p^{0,i}$时，经过S盒的输出为$S(C(i))=q^{3,i}\|q^{2,i}\|q^{1,i}\|q^{0,i}$，其中$0 \leqslant i \leqslant 15$。PICO算法S盒如表1所示。

表1 PICO的S盒

Tab. 1 S-box of PICO

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$\boldsymbol{S}[x]$	1	2	4	d	6	f	b	8	a	5	e	3	9	c	7	0

PICO算法的非线性层由16个相同的S盒并置而成，当S盒的输入为$x=(x_3,x_2,x_1,x_0)$，输出为$y=(y_3,y_2,y_1,y_0)$时，S盒的代数表达式为：

$$\begin{cases} y_0 = 1 + x_0 + x_1 + x_2 + x_3 + x_1x_3 \\ y_1 = x_0 + x_2 + x_3 + x_0x_1 + x_1x_2 + x_1x_2x_3 \\ y_2 = x_1 + x_2 + x_0x_3 + x_2x_3 \\ y_3 = x_3 + x_0x_1 + x_0x_2 + x_0x_3 + x_1x_3 + x_0x_1x_3 \end{cases}$$

3）置换层：PICO算法通过64比特置换表将状态矩阵的第i行第j列的元素$p^{i,j}$移动到表2中对应的位置上，例如$B_P(p^{0,0}) \rightarrow p^{0,10}$，其中置换表如表2所示。

表2 PICO的P盒

Tab. 2 P-box of PICO

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

1.2 PICO算法的密钥扩展方案

PICO的密钥规模为128比特，密钥扩展的设计与SPECK算法类似，主密钥$Key=k^{127}k^{126}\cdots k^1k^0$存储在密钥寄存器中，首先提取子密钥$K^0$和$L^1$，扩展算法可以表示为：

$K^0=k^{63}k^{62}\cdots k^1k^0$

$L^1=k^{127}k^{126}\cdots k^{65}k^{64}$

在得到K^0和L^1后，子密钥K^1到K^{32}的生成过程如下：

Step1 $L_{64}^2=((K_{64}^j)\oplus(L_{64}^1 \gg 3))\oplus(L_{64}^1)$

Step2 $K_{64}^{j+1}=((L_{64}^2)\oplus(K_{64}^j \ll 7))\oplus j$

Step3 $L_{64}^1=L_{64}^2$

其中j从0到31遍历。重复Step1~Step3，便得到了32轮的全部轮密钥。

2 基于比特可分性质的MILP建模

2.1 基于比特的可分性质

对于任意向量$\boldsymbol{a} \in \mathbf{F}_2^n$，$\boldsymbol{a}$的第$i$个元素记为$\boldsymbol{a}[i]$，汉明重量$w(\boldsymbol{a})$定义为$w(\boldsymbol{a})=\sum_{i=0}^{n-1}\boldsymbol{a}[i]$。对任意的$\boldsymbol{a}=(a_0,a_1,\cdots,a_{m-1}) \in (\mathbf{F}_2^{l_0} \times \mathbf{F}_2^{l_1} \times \cdots \times \mathbf{F}_2^{l_{m-1}})$，$\boldsymbol{a}$的汉明重量为$W(\boldsymbol{a})=\left(w(a_0),w(a_1),\cdots,w(a_{m-1})\right) \in \mathbf{Z}^m$。对于$\boldsymbol{k}=(k_0,k_1,\cdots,k_{m-1}) \in \mathbf{Z}^m$和$\boldsymbol{k}'=(k_0',k_1',\cdots,k_{m-1}') \in \mathbf{Z}^m$，若存在

任意的i，都有$k_i \geqslant k'_i$，则称$\boldsymbol{k} \geqslant \boldsymbol{k}'$；否则$\boldsymbol{k} \ngeqslant \boldsymbol{k}'$。

比特乘积函数$\pi_u(x)$和$\pi_U(\boldsymbol{X})$：对于任意$u \in \mathbf{F}_2^n, x \in \mathbf{F}_2^n$，$\pi_u(x)$是满足$u[i]=1$的所有$x[i]$的与，定义为：

$$\pi_u(x) := \prod_{i=1}^{n} x[i]^{u[i]}$$

令$\pi_U(\boldsymbol{X})$是一个$\mathbf{F}_2^{l_0} \times \mathbf{F}_2^{l_1} \times \cdots \times \mathbf{F}_2^{l_{m-1}}$到$\mathbf{F}_2$的函数，对于任意的$\boldsymbol{U}=(x_0, x_1, \cdots, x_{m-1}) \in (\mathbf{F}_2^{l_0} \times \mathbf{F}_2^{l_1} \times \cdots \times \mathbf{F}_2^{l_{m-1}})$，关于输入$\boldsymbol{X}=(x_0, x_1, \cdots, x_{m-1}) \in (\mathbf{F}_2^{l_0} \times \mathbf{F}_2^{l_1} \times \cdots \times \mathbf{F}_2^{l_{m-1}})$的比特乘积函数$\pi_U(\boldsymbol{X})$为：

$$\pi_U(\boldsymbol{X}) := \prod_{i=1}^{n} \pi_{U_i}(\boldsymbol{X}_i)$$

定义 1　可分性[9]。记X为空间$\mathbf{F}_2^{l_0} \times \mathbf{F}_2^{l_1} \times \cdots \times \mathbf{F}_2^{l_{m-1}}$上的多重集，如果集合$\boldsymbol{X}$满足下面的条件称$X$具有可分性$D_K^{l_0, l_1, \cdots, l_{m-1}}$：

$$\bigoplus_{x \in X} \pi_U(\boldsymbol{X}) = \begin{cases} \text{不确定}, & \text{存在} \boldsymbol{k} \in \boldsymbol{K} \text{使得} w(U) \geqslant \boldsymbol{k} \\ 0, & \text{其他} \end{cases}$$

其中：$\boldsymbol{k}$表示一个m维向量；K表示m维向量集合（第i个元素取0到l_i之间的值）。进一步，当考虑比特可分性时，$l_0, l_1, \cdots, l_{m-1}$全被限制为1，可分性质可表示为$D_K^{1^m}$。

定义 2　可分路径[11]。令输入多重集满足可分性$D_{\{\boldsymbol{k}\}}^{l_0, l_1, \cdots, l_{m-1}}$，通过$i$轮加密后，其中间状态满足可分性$D_{K_i}^{l_0, l_1, \cdots, l_{m-1}}$，那么可分性质的传递路径为：

$$\{\boldsymbol{k}\} \triangleq K_0 \rightarrow K_1 \rightarrow \cdots \rightarrow K_r$$

对于任意的向量$\boldsymbol{k}_i \in K_i (i \geqslant 1)$，必定存在向量$\boldsymbol{k}_{i-1} \in K_{i-1}$使得$\boldsymbol{k}_{i-1}$能够传递到$\boldsymbol{k}_i$，其中$i \in \{0, 1, \cdots, r\}$，则存在一条$r$轮的可分路径$(\boldsymbol{k}_0, \boldsymbol{k}_1, \cdots, \boldsymbol{k}_r)$。

2.2　比特可分性的MILP模型

基于MILP搜索比特可分性的核心思想在于将可分性的传递规则转化为一系列线性不等式，通过构建目标函数将积分区分器搜索问题转化为MILP求解问题。下面通过线性不等式对三种基本运算（复制、异或、与）和S盒模型进行描述。

1）模型1（复制）。假设用$(a) \rightarrow (b_0, b_1)$来表示复制操作的一条可分路径，下面的等式能够有效地描述可分性的传递规则：

$$\begin{cases} a - b_0 - b_1 = 0 \\ a, b_0, b_1 \in \{0, 1\} \end{cases}$$

2）模型2（异或）。假设用$(a_0, a_1) \rightarrow (b)$来表示异或操作的一条可分路径，下面的等式能够有效地描述可分性的传递规则：

$$\begin{cases} a_0 + a_1 - b = 0 \\ a_0, a_1, b \in \{0, 1\} \end{cases}$$

3）模型3（与）。假设用$(a_0, a_1) \rightarrow b$来表示与操作的一条可分路径，下面的不等式能够有效地描述可分性的传递规则：

$$\begin{cases} b - a_0 \geqslant 0 \\ b - a_1 \geqslant 0 \\ b - a_0 - a_1 \leqslant 0 \\ a_0, a_1, b \in \{0, 1\} \end{cases}$$

4）S盒模型。为了得到S盒的线性不等式组，首先利用文献[11]提到的算法2得到一个可分路径的完整列表，接下来调用SageMath软件生成刻画S盒的线性不等式集合。通常这个不等式集合规模很大，直接调用Gurobi求解器会使得MILP问题在计算上不可解。为了能够降低计算的复杂度，Sun等[19]提出了贪婪算法（文献[19]中的算法1），调用该算法对S盒的线性不等式集合进行化简，能够显著地降低了S盒不等式的数量。

对于基于复制、异或、与这三种基本操作和S盒的算法，能够构建线性不等式集合来刻画一轮可分性的传递规律。把这种过程迭代r次，将得到一个线性不等式系统来描述r轮可分性质，该系统的所有可行解对应所有r轮可分路径。

初始条件和终止规则：

假设$(a_0^0, a_1^0, \cdots, a_{n-1}^0) \rightarrow \cdots \rightarrow (a_0^r, a_1^r, \cdots, a_{n-1}^r)$表示一条$r$轮的可分路径，$L$是关于$a_i^j (i=0, 1, \cdots, n-1, j=0, 1, \cdots, r)$的线性不等式组。令初始化可分性质为$D_{\{\boldsymbol{k}\}}^{1^n}$，其中$\boldsymbol{k}=(k_0, k_1, \cdots, k_{n-1})$，为了使$L$能够刻画从初始值向量$\boldsymbol{k}$开始的可分路径，需在$L$中添加不等式组$a_i^0 = k_i (i=0, 1, \cdots, n-1)$。为了将$L$完善为MILP问题，还需向$L$中添加目标函数。通过应用可分性的定义，任何汉明重量大于2的向量存在都表明该状态下的所有比特满足零和性质。如果单位向量在输出可分性质中出现，那么说明与该单位向量中非零元素对应的比特位置不遵循零和性质。只需要判断K_r是否包含全部的单位向量，因此设置目标函数为：

$$Obj: \text{Min}\{a_0^r + a_1^r + \cdots + a_{n-1}^r\}$$

通过设置限制条件L和目标函数Obj便得到了一个完整的MILP模型，令$D_{K_i}^{1^n}$表示i轮加密后的输出可分性质，当K_{r+1}首次包含所有n个单位向量，可分性质停止传递，由$D_{K_r}^{1^n}$搜索得到一个r轮的积分区分器。

3　PICO算法的建模过程

3.1　比特可分性的MILP模型

PICO算法采用SPN结构，轮密钥加操作并不影响可分性质的传递，因此不将其考量在本文的分析当中。首先应用S盒传递规则，计算得到通过每个S盒的49条可分路径，如表3所示。

表3　PICO S盒的可分路径

Tab. 3　Division paths of PICO S-box

输入$D_k^{1,4}$	输出$D_K^{1,4}$
(0,0,0,0)	(0,0,0,0)
(0,0,0,1)	(0,0,0,1),(0,0,1,0),(0,1,0,0),(1,0,0,0)
(0,0,1,0)	(0,0,0,1),(0,0,1,0),(0,1,0,0),(1,0,0,0)
(0,0,1,1)	(0,0,1,0),(0,1,0,1),(1,0,0,0)
(0,1,0,0)	(0,0,0,1),(0,0,1,0),(0,1,0,0),(1,0,0,0)
(0,1,0,1)	(0,0,1,0),(0,1,0,1),(1,0,0,0)
(0,1,1,0)	(0,0,1,0),(0,1,0,1),(1,0,0,0)
(0,1,1,1)	(0,1,1,1),(1,0,0,1),(1,1,1,0)
(1,0,0,0)	(0,0,0,1),(0,0,1,0),(0,1,0,0),(1,0,0,0)
(1,0,0,1)	(0,0,1,1),(0,1,0,0),(1,0,0,0)
(1,0,1,0)	(0,0,0,1),(0,0,1,0),(1,0,0,0)
(1,0,1,1)	(0,0,1,1),(0,1,1,0),(1,0,0,0)
(1,1,0,0)	(0,0,1,0),(0,1,0,0),(1,0,0,1),(0,0,1,1)
(1,1,0,1)	(0,0,1,1),(1,1,0,1),(1,1,1,0)
(1,1,1,0)	(0,0,1,0),(0,1,0,1),(1,1,0,0)
(1,1,1,1)	(1,1,1,1)

其中$(a_3, a_2, a_1, a_0) \rightarrow (b_3, b_2, b_1, b_0)$表示一条可分路径，这49条可分路径形成了一个点集，并将点集输入到SageMath

软件中的inequality_generator()函数中，将返回52个线性不等式集合，再利用贪婪算法化简得到约束每个S盒的15个线性不等式组L，表示如下：

$$L=\begin{cases}a_3+a_2+a_1+a_0-b_3-b_2-b_1-b_0\geqslant 0\\ -a_3-2a_2-2a_1-3a_0+4b_3+2b_2+3b_1+b_0+2\geqslant 0\\ -2a_3-2a_2-a_1-3b_3+2b_2-b_1+b_0+6\geqslant 0\\ 3a_2-2b_3-b_2-b_1-b_0+2\geqslant 0\\ -a_3+a_1+a_0+b_3-2b_2-2b_1-b_0+3\geqslant 0\\ 2a_3+a_2+a_1+a_0-2b_3-2b_1-2b_0+1\geqslant 0\\ -a_3-2a_2-2a_1-3a_0+b_3-b_2+2b_1+b_0+5\geqslant 0\\ -a_3-a_2-2a_1+3b_3+2b_2+4b_1+3b_0\geqslant 0\\ a_3+a_2+a_1+3a_0-2b_3-2b_2-2b_1-b_0+1\geqslant 0\\ -2a_3+2a_2-a_1-b_3-2b_2+b_1-b_0+4\geqslant 0\\ a_3+a_0-b_3+b_2-2b_1-2b_0+2\geqslant 0\\ a_3+b_3-b_2-b_1+b_0+1\geqslant 0\\ -2a_3-3a_2-a_1-a_0+b_3+2b_2+3b_1+b_0+3\geqslant 0\\ -a_2-a_0-b_2+b_1+b_0+2\geqslant 0\\ -a_3-a_0+b_2-b_1+b_0+2\geqslant 0\end{cases}$$

因此PICO算法的非线性层可用15 × 16 = 240个线性不等式表示。

PICO算法的线性层通过64比特置换表进行比特移位，因此只需在下一轮可分性质传递过程中改变向量系数的位置。有了描述S层和线性层的不等式组，便得到了一轮PICO算法可分性传递的不等式组。迭代r轮之后，便可构造r轮具有可分性路径的线性不等式组，将其作为MILP模型的限制条件。只要给定初始可分性，便可求解该MILP模型是否存在积分区分器。

3.2 PICO算法的积分区分器

本节根据3.1节建立的PICO算法的MILP模型，采用python编程建模求解，首次给出了PICO算法的两个积分区分器。首先根据活跃比特数越多、搜索轮数越长的性质，设定活跃比特数为63，通过大量实验搜索得到PICO算法11个10轮积分区分器，这也是目前为止PICO算法已知最长的积分区分器。但有些区分器平衡比特数太少，区分优势不明显，通过筛选给出了PICO算法输入63个活跃比特、输出26个比特平衡的10轮积分区分器。假设a表示活跃，b表示平衡，c表示常数，“?”表示未知，目前搜索到PICO算法最长的10轮积分区分器输入状态可表示为：

aaaa, aaaa, aaaa, aaaa, aaaa, aaaa, aaaa, aaaa

acaa, aaaa, aaaa, aaaa, aaaa, aaaa, aaaa, aaaa

输出可表示为：

????,b???,b?b?,b?b?,b?b?,b?b?,b???,b???

b?b?,b?b?,??b?,????,b?b?,b?b?,bbb?,bbb?

但由于10轮积分区分器输入活跃比特数较多，最多可利用2组明文，不利于密钥恢复；同时，活跃比特数越多，意味着数据复杂度越大。因此，通过将对应于S盒的连续4位设置为常数值，其余位置设置为活跃，遍历所有的情况，再次搜索得到PICO算法13个9轮积分区分器。根据平衡比特数越多，区分优势越明显的性质。本文选择如下的9轮积分区分器，输入60个活跃比特，输出33个平衡比特，并根据此区分器对PICO算法进行11轮积分攻击。9轮积分区分器输入状态可表示为：

aaaa, aaaa, aaaa, aaaa, aaaa, aaaa, aaaa, cccc

aaaa, aaaa, aaaa, aaaa, aaaa, aaaa, aaaa, aaaa

输出可表示为：

b?b?,b???, b?b?, b???, b?b?, b?b?, b?b?, b?b?

bbbb,b?b?,b?b?,b?b?, b?b?,b?b?,bbb?,b?b?

4 PICO算法的密钥恢复攻击

利用3.2节给出的9轮积分区分器，向后攻击两轮，对PICO算法进行11轮积分攻击。首先，通过选取特定构造9轮区分器所需要的明文，得到11轮加密后的密文。然后，以平衡位置所在的列为单位进行筛选。每一次筛选中，需要猜测第11轮的轮密钥RK^{11}及第10轮的轮密钥RK^{10}的部分密钥信息，其中已猜测的部分密钥信息在之后的筛选过程中默认已经唯一确定。最后，对密文进行解密恢复出第9轮的结果，通过验证第9轮输出的对应位置比特是否平衡来筛选出正确密钥。

平衡位置的选择不同时，对应筛选出RK^{11}和RK^{10}的密钥字(4比特)也不同。通过16次筛选后，便可得到平衡位置与第11轮的轮密钥密钥RK^{11}及第10轮轮密钥RK^{10}之间的关系，具体如表4所示。

表4 平衡位置与可筛选密钥字的对应关系

Tab. 4 Relationship between balanced positions and nibbles of recovered round keys

序号	区分器平衡位置	RK^{10}	RK^{11}	使用11组明文剩余错误密钥期望值$\overline{N}$
1	32,33,34,35	7	**12,2,1**	$(2^{16}-1)\times(2^{-4})^{11}\approx 2^{-28}$
2	56,57,58	1	**8**,**7**,**5**,2	$(2^{16}-1)\times(2^{-3})^{11}\approx 2^{-15}$
3	0,2	15	**15**,**9**,**6**,**3**	$(2^{20}-1)\times(2^{-2})^{11}\approx 2^{-2}$
4	8,10	13	12,**11**,**10**,**4**	$(2^{16}-1)\times(2^{-2})^{11}\approx 2^{-6}$
5	16,18	11	15,10,6,3	$(2^{4}-1)\times(2^{-2})^{11}\approx 2^{-18}$
6	20,22	10	**14**,10,7,6	$(2^{8}-1)\times(2^{-2})^{11}\approx 2^{-14}$
7	24,26	9	**13**,10,7,4	$(2^{8}-1)\times(2^{-2})^{11}\approx 2^{-14}$
8	28,30	8	14,**0**	$(2^{8}-1)\times(2^{-2})^{11}\approx 2^{-14}$
9	36,38	6	14,13,3,1	$(2^{4}-1)\times(2^{-2})^{11}\approx 2^{-18}$
10	40,42	5	9,8,5,1	$(2^{4}-1)\times(2^{-2})^{11}\approx 2^{-18}$
11	44,46	4	15,11,9,4	$(2^{4}-1)\times(2^{-2})^{11}\approx 2^{-18}$
12	48,50	3	13,8,3,2	$(2^{4}-1)\times(2^{-2})^{11}\approx 2^{-18}$
13	52,54	2	14,13,11	$(2^{4}-1)\times(2^{-2})^{11}\approx 2^{-18}$
14	60,62	0	12,7,6,5	$(2^{4}-1)\times(2^{-2})^{11}\approx 2^{-18}$
15	4	14	10,8,7,5	$(2^{4}-1)\times(2^{-1})^{11}\approx 2^{-7}$
16	12	12	12,6,4,2	$(2^{4}-1)\times(2^{-1})^{11}\approx 2^{-7}$

本文选择32、33、34、35这四个平衡位置为例进行说明，攻击流程如图2所示。图2内部结构与1.1节给出PICO算法的状态矩阵相对应，该状态矩阵的每一列都可以表示成一个向量，即可以表示成$\boldsymbol{Q}_{15}$、$\boldsymbol{Q}_{14}$、…、$\boldsymbol{Q}_0$共16个向量，其中$\boldsymbol{Q}_j=(p^{0,j},p^{1,j},p^{2,j},p^{3,j})^{\mathrm{T}}$，$0\leqslant j\leqslant 15$。下面给出攻击的具体步骤：

Step1 根据输入的初始状态构造9轮积分区分器，由于输入状态有4比特为常数比特，因此将$p^{0,8}$、$p^{1,8}$、$p^{2,8}$、$p^{3,8}$对应位置取定为常数，其余60个活跃位置遍历$\{0,1\}^{60}$，故每组包含2^{60}个明文，并对其进行11轮加密，相应的密文记为C_0、C_1、…、$C_{2^{60}-1}$。

Step2 通过猜测密钥RK^{11}的3个密钥字RK_{12}^{11}、RK_2^{11}、RK_1^{11}共12比特密钥信息，对2^{60}个密文进行第11轮解密，计算得到第10轮的12比特输出，其中计算表达式为$V_j^{(i)}=S^{-1}(P^{-1}(C_i))\oplus RK_j^{11}$，$j\in\{1,2,12\}$。

Step3 计算第10轮通过S盒后的中间状态，即$T^i=P^{-1}(V^{(i)})$，猜测密钥RK^{10}的一个密钥字RK_7^{10}，计算$t_i=$

$S^{-1}(T_7^i)\oplus RK_7^{10}$。

Step4　若猜测密钥RK_{12}^{11}、RK_2^{11}、RK_1^{11}、RK_7^{10}正确,一定能保证$\bigoplus_{0\leqslant i\leqslant 2^{60}-1} t_i$的第32~35比特均为0,将所猜测的密钥保留为候选密钥;否则,淘汰错误密钥。

Step5:　选择另一组构造9轮区分器时的明文,重复Step1~Step4,直到唯一确定RK_{12}^{11}、RK_2^{11}、RK_1^{11}、RK_7^{10}。

复杂度分析:表4利用区分器平衡位置由多到少依次对密钥字进行筛选,即攻击时共需要筛选16次。RK^{11}已确定的密钥字在表4中用粗体标注出来,在下次筛选时无需猜测已确定的密钥字。为了唯一确定正确密钥,即保证每一种情况下错误密钥的期望值$\overline{N}$都小于1,因此需要选择11组明文进行分析。经过16次筛选后,RK^{10}的16个子密钥和RK^{11}的16个子密钥共128比特密钥信息将唯一确定。从而攻击的数据复杂度为11组($2^{60}\times 11\approx 2^{63.46}$)明文,低于明文直接穷举量。每次筛选过程中猜测密钥字的个数不同,其中需猜测5个密钥字的1次、4个密钥字的3次、2个密钥字的3次、1个密钥字的9次,所以处理11组明文整个攻击的时间复杂度共约需$2^{63.46}\times(2^{4\times 5}+3\times 2^{4\times 4}+3\times 2^{4\times 2}+9\times 2^{4\times 1})\approx 2^{83.46}$次S盒查表,这相当于$2^{83.46}/(11\times 16)\approx 2^{76}$次11轮加密。此外,为了对猜测的密钥字进行存储,存储复杂度为$2^{4\times 5}+3\times 2^{4\times 4}+3\times 2^{4\times 2}+9\times 2^{4\times 1}\approx 2^{20}$。

	0	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60
	1	5	9	13	17	21	25	29	33	37	41	45	49	53	57	61
	2	6	10	14	18	22	26	30	34	38	42	46	50	54	58	62
Round 9	3	7	11	15	19	23	27	31	35	39	43	47	51	55	59	63
Round10	0	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60
	1	5	9	13	17	21	25	29	33	37	41	45	49	53	57	61
RK^{10}, S	2	6	10	14	18	22	26	30	34	38	42	46	50	54	58	62
	3	7	11	15	19	23	27	31	35	39	43	47	51	55	59	63
	45	29	54	15	52	26	0	51	56	19	63	14	38	13	41	20
	3	30	50	33	44	16	47	5	6	12	43	46	1	57	36	31
P	21	53	24	60	55	11	22	42	25	61	59	27	18	48	34	28
	17	37	39	8	10	7	40	58	62	2	4	9	49	32	35	23
Round11	0	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60
	1	5	9	13	17	21	25	29	33	37	41	45	49	53	57	61
RK^{11}, S	2	6	10	14	18	22	26	30	34	38	42	46	50	54	58	62
	3	7	11	15	19	23	27	31	35	39	43	47	51	55	59	63
	45	29	54	15	52	26	0	51	56	19	63	14	38	13	41	20
	3	30	50	33	44	16	47	5	6	12	43	46	1	57	36	31
P	21	53	24	60	55	11	22	42	25	61	59	27	18	48	34	28
	17	37	39	8	10	7	40	58	62	2	4	9	49	32	35	23

图2　11轮PICO的积分攻击

Fig. 2　Integral attack on 11-round PICO

下面将本文积分攻击与零相关线性分析方法进行比较,结果如表5所示。

表5　本文方法与零相关攻击方法的实验结果对比

Tab. 5　Experimental results comparison of proposed method and zero correlation attack method

攻击方法	区分器长度/轮	攻击长度/轮	恢复密钥量/b	数据复杂度	时间复杂度	存储复杂度
零相关攻击[15]	7	10	50	$2^{63.3}$	$2^{68.7}$次10轮加密	$2^{42.3}$
本文积分攻击	9	11	128	$2^{63.46}$	2^{76}次11轮加密	2^{20}

5　结语

PICO算法能够很好地抵抗差分攻击[20]、线性攻击[21]、biclique攻击[22]、零相关攻击[23]和相关密钥攻击[24],但迄今为止未有人对PICO算法抵抗积分攻击的能力进行研究。本文主要采用基于比特可分性的MILP建模方法对PICO算法进行积分分析,得到了轮数最长的10轮积分区分器,但由于活跃比特数太多,可利用的明文数太少,不利于密钥恢复。本文选择搜索到的9轮积分区分器进行11轮的积分攻击,该攻击经过16次的筛选能够恢复PICO算法128比特密钥信息。其中攻击数据复杂度为$2^{63.46}$,时间复杂度为2^{76}次11轮加密,存储复杂度为2^{20},本文首次给出攻击11轮PICO算法的数据复杂度小于穷举攻击的有效攻击。分析结果表明了11轮PICO算法不能抵抗积分攻击,但由于在搜索中活跃比特数越多,搜索得到的积分区分器越长,因此使得选择明文的数据量大大增加,如何能够平衡活跃比特数和长轮数之间的关系将是进一步需要解决的问题;同时MILP搜索积分区分器的方法还有一定的局限性,采用文献[11]方法产生的不等式不足以约束大规模的S盒,如8进8出的S盒,解决S盒算法的适应性问题也将是下一步研究的主要方向。

参考文献 (References)

[1] BANIK S, PANDEY S K, PEYRIN T, et al. GIFT: a small present [C]// Proceedings of the 2017 International Conference on Cryptographic Hardware and Embedded Systems, LNCS 10529. Cham: Springer, 2017: 321-345.

[2] BANIK S, BOGDANOV A, ISOBE T, et al. Midori: a block cipher for low energy [C]// Proceedings of the 2015 International Conference on Cryptology and Information Security, LNCS 9453. Berlin: Springer, 2015: 411-436.

[3] GUO J, PEYRIN T, POSCHMANN A, et al. The LED block cipher [C]// Proceedings of the 2011 International Workshop on Cryptographic Hardware and Embedded Systems, LNCS 6917. Berlin: Springer, 2011: 326-341.

[4] BEIERLE C, JEAN J, KÖLBL S, et al. The SKINNY family of block ciphers and its low-latency variant MANTIS [C]// Proceedings of the 2016 Annual International Cryptology Conference, LNCS 9815. Berlin: Springer, 2016: 123-153.

[5] DAEMEN J, KNUDSEN L, RIJMEN V. The block cipher Square [C]// Proceedings of the 1997 International Workshop on Fast Software Encryption, LNCS 1267. Berlin: Springer, 1997: 149-165.

[6] BIRYUKOV A, SHAMIR A. Structural cryptanalysis of SASAS [C]// Proceedings of the 2001 International Conference on the Theory and Applications of Cryptographic Techniques, LNCS 2045. Berlin: Springer, 2001: 395-405.

[7] LUCKS S. The saturation attack — a bait for Twofish [C]// Proceedings of the 2001 International Workshop on Fast Software Encryption, LNCS 2355. Berlin: Springer, 2001: 1-15.

[8] Z'ABA M R, RADDUM H, HENRICKSEN M, et al. Bit-pattern based integral attack [C]// Proceedings of the 2008 International Workshop on Fast Software Encryption, LNCS 5086. Berlin: Springer, 2008: 363-381.

[9] TODO Y. Structural evaluation by generalized integral property

[C]// Proceedings of the 2015 Annual International Conference on the Theory and Applications of Cryptographic Techniques. LNCS 9056. Berlin: Springer, 2015: 287-314.

[10] TODO Y. Integral cryptanalysis on full MISTY11 [C]// Proceedings of the 2008 International Workshop on 2015 Annual Cryptology Conference, LNCS 9215. Berlin: Springer, 2015: 413-432.

[11] XIANG Z, ZHANG W, BAO Z, et al. Applying MILP method to searching integral distinguishers based on division property for 6 lightweight block ciphers [C]// Proceedings of the 2016 International Conference on the Theory and Application of Cryptology and Information Security, LNC 10031. Berlin: Springer, 2016: 648-678.

[12] 信文倩,孙兵,李超. LiCi算法的基于比特积分攻击[J/OL]. 计算机工程 [2020-03-27]. http://kns. cnki. net/kcms/detail/31. 1289. TP. 20190828. 1759. 006. html. (XIN W Q, SUN B, LI C. Bit-based integral attack on LiCi [J/OL]. Journal of Computer Engineering [2020-03-27]. http://kns. cnki. net/kcms/detail/31. 1289. TP. 20190828. 1759. 006. html)

[13] 尚方舟,沈璇,刘国强,等. 基于MILP搜索的PUFFIN算法积分分析[J]. 密码学报, 2019, 6(5): 627-638. (SHANG F Z, SHEN X, LIU G Q, et al. Integral cryptanalysis on PUFFIN based on MILP [J]. Journal of Cryptologic Research, 2019, 6(5): 627-638.)

[14] 李艳俊,梁萌. 基于比特可分性的BORON和Khudra积分区分器搜索[J/OL]. 计算机应用研究 [2020-03-27]. https://kns. cnki. net/KCMS/detail/51. 1196. TP. 20191024. 1009. 031. html. (LI Y J, LIANG M. Integral distinguisher search of BORON and Khudra based on bit-based division property [J] Application Research of Computers [2020-03-27]. https://kns. cnki. net/KCMS/detail/51. 1196. TP. 20191024. 1009. 031. html.)

[15] 马楚焱,刘国强,李超. 对PICO和RECTANGLE的零相关线性分析[J]. 密码学报, 2017, 4(5): 413-422. (MA C Y, LIU G Q, LI C. Zero-correlation linear cryptanalysis on PICO and RECTANGLE[J]. Journal of Cryptologic Research, 2017, 4(5): 413-422.)

[16] 马楚焱. 混合整数线性规划在分组密码安全性分析中的应用[D]. 长沙:国防科技大学, 2017. (MA C Y. Application of mixed integer linear programming in block cipher security analysis [D]. Changsha: National University of Defense Technology, 2017.)

[17] BOGDANOV A, LEANDER G, NYBERG K, et al. Integral and multidimensional linear distinguishers with correlation zero [C]// Proceedings of the 2012 International Conference on the Theory and Application of Cryptology and Information Security, LNCS 7658. Berlin: Springer, 2012: 244-261.

[18] BANSOD G, PISHAROTY N, PATIL A. PICO: an ultra lightweight and low power encryption design for ubiquitous computing[J]. Defence Science Journal, 2016, 66(3): 259-265.

[19] SUN S, HU L, WANG P, et al. Automatic security evaluation and (related-key) differential characteristic search: application to SIMON, PRESENT, LBlock, DES (L) and other bit-oriented block ciphers [C]// Proceedings of the 2014 International Conference on the Theory and Application of Cryptology and Information Security, LNCS 8873. Berlin: Springer, 2014: 158-178.

[20] BIHAM E, SHAMIR A. Differential cryptanalysis of DES-like cryptosystems[J]. Journal of Cryptology, 1991, 4(1): 3-72.

[21] MATSUI M. Linear cryptanalysis method for DES cipher [C]// Proceedings of the 1993 Workshop on the Theory and Application of Cryptographic Techniques, LNCS 765. Berlin: Springer, 1993: 386-397.

[22] BOGDANOV A, KHOVRATOVICH D, RECHBERGER C. Biclique cryptanalysis of the full AES [C]// Proceedings of the 2011 International Conference on the Theory and Application of Cryptology and Information Security, LNCS 7073. Berlin: Springer, 2011: 344-371.

[23] BOGDANOV A, RIJMEN V. Linear hulls with correlation zero and linear cryptanalysis of block ciphers[J]. Designs, Codes and Cryptography, 2014, 70(3): 369-383.

[24] BIHAM E. New types of cryptanalytic attacks using related keys [J]. Journal of Cryptology, 1994, 7(4): 229-246.

This work is partially supported by the Cryptology Theory Project of National Cryptology Development Fund During the 13th Five Year Plan (MMJJ20180217).

LIU Zongfu, born in 1995, M. S. candidate. His research interests include security analysis of symmetric cryptographic algorithm.

YUAN Zheng, born in 1968, Ph. D., professor. Her research interests include cipher design, cryptanalysis, obfuscation in cryptography.

ZHAO Chenxi, born in 1996, M. S. candidate. Her research interests include security analysis of symmetric cryptographic algorithm.

ZHU Liang, born in 1995, M. S. candidate. His research interests include security analysis of symmetric cryptographic algorithm.