基于MILP对轻量级密码算法FBC-128的差分分析

赵 琪,樊 婷,韦永壮
(桂林电子科技大学广西密码学与信息安全重点实验室,广西桂林 541004)

摘 要: FBC(Feistel-based Block Cipher)是入围全国密码算法设计竞赛第二轮的轻量级分组密码.由于它具备算法结构简洁、安全性高及软硬件实现性能卓越等优点,备受业界广泛关注.FBC密码算法的数据分组长度和密钥长度至少为128比特,记为FBC-128.目前对FBC-128算法差分攻击的最好结果是12轮,时间复杂度为$2^{93.41}$次加密,数据复杂度为2^{122}个选择明文对.然而,FBC算法是否存在更长的差分区分器,能否对其进行更高轮数的密钥恢复攻击仍有待解决.本文基于混合整数线性规划(MILP)的自动化搜索方法,提出了"分段统计法"来求解FBC-128的差分特征.实验测试结果表明:FBC-128存在15轮差分区分器,其概率为2^{-121}.然后将其向后扩展1轮,对16轮FBC-128算法发起密钥恢复攻击,其数据复杂度为2^{121}个选择明文数据量,时间复杂度为$2^{92.68}$次加密.与已有结果相比,差分区分器和密钥恢复攻击都提升了4轮,并且所需的数据复杂度和时间复杂度更低.

关键词: 自动化分析;混合整数线性规划;分组密码算法;差分区分器;密钥恢复攻击;FBC算法
基金项目: 国家自然科学基金(No.62162016);广西自然科学基金创新研究团队项目(No.2019GXNSFGA245004)
中图分类号: TN918 **文献标识码:** A **文章编号:** 0372-2112(2024)06-1896-07
电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20230161

MILP-Based Differential Cryptanalysis of the FBC-128 Lightweight Cipher

ZHAO Qi, FAN Ting, WEI Yong-zhuang
(*Guangxi Key Laboratory of Cryptography and Information Security*, *Guilin University of Electronic Technology*, *Guilin*, *Guangxi* 541004, *China*)

Abstract: FBC (Feistel-based Block Cipher) is a lightweight block cipher selected in the second round of the National Cryptographic Algorithm Design Competition. It has many advantages such as simple algorithm structure, high security and excellent implementation performance, and has attracted much attention in the industry. The block size and key length of FBC are at least 128 bits, denoted as FBC-128. At present, the best result of differential attack on FBC-128 is 12-round. The time complexity is $2^{93.41}$ encryptions, and the data complexity is 2^{122} chosen-plaintexts. However, it is still to be solved whether there is a longer differential distinguisher and higher rounds of key recovery attack on FBC. In this paper, a segmental statistical method is proposed to search the differential characteristic of FBC-128 based on the mixed-integer linear programming technology. The results show that FBC-128 exists15-round differential distinguisher with probability 2^{-121}. Then, we extend it backward by one round, and launch a key recovery attack on 16-round FBC-128.The data complexity is 2^{121} chosen-plaintexts, and the time complexity is $2^{92.68}$ encryptions. Compared with the existing results, the differential distinguisher and key recovery attacks are increased by 4 rounds with lower data and time complexity.

Key words: automatic analysis; mixed-integer linear programming; block cipher; differential distinguisher; key recovery attack; FBC cipher
Foundation Item(s): National Natural Science Foundation of China (No.62162016); The Innovation Research Team Project of Guangxi Natural Science Foundation (No.2019GXNSFGA245004)

收稿日期:2023-02-24;修回日期:2023-10-25;责任编辑:李勇锋

1　引言

随着6G通信网络技术的发展,物联网设备在各行业实现了广泛的应用. 许多物联网设备的应用场景为资源受限的环境,而传统的分组密码通常软、硬件实现要求较高,无法满足资源受限设备的数据安全需要,轻量级分组密码算法应运而生. 与此同时,为防范信息可能被泄露的风险,轻量级密码算法在保证实现效率的同时,也要达到足够的安全强度来抵御现有的攻击. 1991年,Biham等人[1]针对DES[2]类密码算法提出差分分析方法,通过利用高概率的差分特征来恢复密钥. 随后差分密码分析被广泛应用于密码分析中,成为最基本的密码分析方法之一,同时也是衡量密码算法安全性的重要指标. 差分分析主要分为两步:一是寻找到有效的差分区分器;二是利用所找到的区分器来进行密钥恢复攻击. 因此,差分分析的关键在于找到一条高概率且覆盖轮数较长的差分特征.

近年来,由于自动化分析方法具有精确度高、操作简易,效率较高等优点,基于混合整数线性规划(Mixed Integer Linear Programming, MILP)的自动化搜索技术成为密码安全性分析领域应用最广泛的工具之一. 2011年,Mouha等人[3]首次将MILP方法应用于求解差分(线性)活跃S盒数量下界,以此评估密码算法的安全性. 2014年,孙思维等人将MILP方法推广至面向比特的密码算法的安全性评估工作[4],并引入概率变量来寻找最优差分特征[5]. 2017年,为了精简S盒的差分传播模型,Sasaki等人[6]提出利用MILP方法来进行约简,用更少的不等式来描述S盒的差分传播过程. 2019年,基于面向比特的MILP自动化方法,Zhu等人[7]提出了分割方法来搜索长轮的GIFT-128算法[8]差分特征. 在FSE2020会议上,Boura等人[9]提出了新的方法来对复杂的线性层和S盒进行建模,显著减少了一些算法部件建模所需不等式数量. 2021年,Zong等人[10]改进了MILP模型,对有利于密钥恢复的区分器进行搜索,以便发起更高轮的密钥恢复攻击. 2022年,Makarim等人[11]使用可满足性模理论(Satisfiability Modulo Theories, SMT)[12]与MILP结合的混合搜索策略得到了Ascon算法[13]更严格的差分界. Li等人[14]提出"超级球"方法,实现小状态S盒的高效刻画. 经过数年的发展和完善,基于MILP的自动化工具在密码分析领域应用广泛.

2018年,美国国家标准与技术研究院发起了轻量级密码算法标准征集竞赛,经过了四年的研究讨论,最终选择Ascon作为轻量级算法标准. 同样在2018年,中国密码学会举办了全国密码算法设计竞赛,最终22个候选分组密码算法中有十个算法脱颖而出,进入第二轮评选,其中包括轻量级分组密码算法FBC[15]. 由于FBC算法结构简洁,实现高效,安全性高,受到了广泛关注. 算法设计者借助MILP工具证明其可以抵抗差分、线性、不可能差分以及积分分析. 2019年,Ren等人[16]采用分支定界方法[17]找到FBC-128算法的11轮差分区分器,并发起了12轮密钥恢复攻击,其时间复杂度和数据复杂度分别为$2^{93.41}$和2^{122}. 2022年,张毅[18]等人寻找到FBC-128算法9轮截断不可能差分,并发起了13轮的不可能差分攻击. 是否可以寻找到FBC算法更高轮的差分区分器,进行更高轮的密钥恢复攻击,仍然亟待进一步的研究.

为提高差分特征的搜索效率,本文采用"分段统计法"对长轮差分特征分段进行搜索. 最终我们寻找到15轮FBC-128算法差分区分器,概率为2^{-121}. 基于该差分区分器向后扩展一轮,发起了FBC-128算法的16轮密钥恢复攻击,时间复杂度为$2^{92.68}$次16轮加密,数据复杂度为2^{121}个明文对. 相比于文献[16]中的结果,在时间复杂度和数据复杂度更低的情况下,将攻击的轮数提高了四轮. 目前对FBC-128算法的分析结果如表1所示.

表1　FBC-128算法分析结果

分析方法	区分器轮数	攻击轮数	时间复杂度	数据复杂度	文献
差分分析	11	12	$2^{93.41}$	2^{122}	[16]
线性分析	10	11	$2^{122.54}$	2^{84}	[16]
不可能差分分析	7	11	$2^{94.54}$	2^{127}	[16]
不可能差分分析	9	13	$2^{122.96}$	2^{126}	[18]
差分分析	15	16	$2^{92.68}$	2^{121}	本文

2　预备知识

2.1　符号说明

本文涉及到的符号、运算符含义如下所示.

R: 加密轮数;

n: 分组大小;

m: 主密钥长度;

w: 字宽度;

F_i: 第i个轮函数,$0 \leqslant i \leqslant 2R-1$;

K_i: 第i个轮密钥,$0 \leqslant i \leqslant 2R-1$;

X_j^i: 第i轮的第j个字,$0 \leqslant i \leqslant R, 0 \leqslant j \leqslant 4$;

S: S盒操作;

$\lll$: 循环左移;

$\|$: 连接操作;

$\oplus$: 异或操作;

$\&$: 与操作.

2.2 FBC算法简介

FBC算法主要包含三个版本，分别是FBC128-128，FBC128-256，FBC256-256. 本文主要关注分组大小和密钥长度均为128比特的FBC算法版本，记为FBC-128. 各版本的主要参数及迭代轮数如表2所示.

表2 FBC算法参数

版本	分组长度n	主密钥长度m	字宽度w	迭代轮数R
FBC128-128	128	128	32	48
FBC128-256	128	256	32	64
FBC256-256	256	256	64	80

（1） 算法结构

FBC算法采用了4路两重Feistel结构，算法主体结构如图1所示.

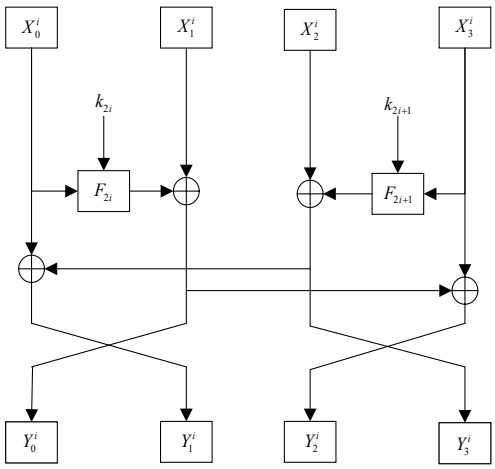

图1 FBC算法结构图

考虑第$i(0\leqslant i\leqslant R)$轮的加密过程，$(X_0^i,X_1^i,X_2^i,X_3^i)\in(\mathbb{F}_2^w)^4$和$(Y_0^i,Y_1^i,Y_2^i,Y_3^i)\in(\mathbb{F}_2^w)^4$分别表示第$i$轮的输入和输出状态，$F_{2i}$和$F_{2i+1}$表示第$i$轮的轮函数，$k_{2i}\in\mathbb{F}_2^w$和$k_{2i+1}\in\mathbb{F}_2^w$表示第$i$轮的轮密钥，则第$i$轮的加密过程可表示如下：

$$\begin{cases}Y_0^i=F_{2i}(X_0^i,k_{2i})\oplus X_1^i\\ Y_1^i=F_{2i+1}(X_3^i,k_{2i+1})\oplus X_2^i\oplus X_0^i\\ Y_2^i=F_{2i}(X_0^i,k_{2i})\oplus X_1^i\oplus X_3^i\\ Y_3^i=F_{2i+1}(X_3^i,k_{2i+1})\oplus X_2^i\end{cases}\tag{1}$$

（2） 轮函数F

FBC算法的轮函数F主要由子密钥加、列变换、行变换三个步骤组成.

（a）子密钥加：子密钥异或轮函数的输入状态；

（b）列变换：记u和v分别为列变换操作的输入和输出，首先将其按照字长w分为四部分，即$u=u_0\|u_1\|u_2\|u_3$，$v=v_0\|v_1\|v_2\|v_3$，随后进行以下操作：

$$v_0[j]\|v_1[j]\|v_2[j]\|v_3[j]=S(u_0[j]\|u_1[j]\|u_2[j]\|u_3[j])\tag{2}$$

其中，$0\leqslant j\leqslant w/4-1$，$S$为4比特S盒代换，如表3所示.

表3 FBC算法的S盒代换（十六进制）

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

（c）行变换：$L_{s,t}(v)=v\oplus(v\lll s)\oplus(v\lll t)$，其中$s,t$的取值与字长$w$有关，FBC-128版本中$s,t$分别取3和10.

由于本文只考虑单密钥情况下FBC-128密码算法的差分分析，故对密钥扩展算法不再赘述. 关于FBC算法的详细信息，可参考文献[15].

3 基于MILP的FBC单密钥差分搜索模型

3.1 面向比特的MILP自动化搜索方法

混合整数线性规划是一类线性规划问题，主要求解给定线性约束条件下目标函数的最大值或最小值，并且规定部分或全部变量为整数值. 近年来，MILP技术被广泛应用于密码算法的安全性评估，其中心思想是通过不等式来刻画密码算法的各个部件密码特性，建立MILP模型. 例如在差分分析中，需要对算法的每个部件构建差分传播模式，通过搜索最小活跃S盒数或者高概率差分特征，评估密码算法是否能抵抗差分分析，发起相应的密钥恢复攻击. 本文基于FBC算法的结构，构建了面向比特的MILP模型，用于搜索FBC算法的差分区分器，与面向字的自动化搜索模型相比，结果更加精确、有效. 以下详细介绍本文中FBC算法各个部件差分传播模式的不等式建模方法.

（1） 异或操作

假设异或操作运算表示为：$c=a\oplus b$，使用以下4个不等式来描述异或操作的差分传播模型，相比于文献[4]中所使用的建模方法，减少了一个不等式和一个虚拟变量.

$$\begin{cases}a+b+c-2\leqslant 0\\ a-b-c\leqslant 0\\ b-a-c\leqslant 0\\ a+b-c\geqslant 0\end{cases}\tag{3}$$

（2） S盒

FBC算法的非线性部件使用一个4比特S盒，其差分分布表（Differential Distribution Table，DDT）如表4所示.

要对S盒的差分传播过程进行描述，需要对DDT表中所有可能的差分传播模式进行刻画，本文主要采用文献[4]中凸包建模方法得到不等式模型. 输入差分和输出差分可表示为(x_0,x_1,x_2,x_3)和(y_0,y_1,y_2,y_3)，其中x_i,y_i均为0-1变量（$0\leqslant i,j\leqslant 4$）. 根据DDT表，可能的非0概率只有三种：$2^0,2^{-2},2^{-3}$，因此可引入两个0-1变量

表4　FBC算法S盒差分分布表

	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	2	0	4	0	2	0	0	0	2	0	0	0	2
2	0	0	4	0	0	0	4	0	0	0	4	0	0	0	4	0
3	0	4	0	2	0	4	0	2	0	0	0	2	0	0	0	2
4	0	4	0	0	4	0	0	0	0	0	0	0	4	4	0	0
5	0	0	0	2	4	0	0	2	0	0	0	2	4	0	0	2
6	0	0	2	2	0	0	2	2	0	0	2	2	0	0	2	2
7	0	4	2	0	0	0	2	0	0	0	2	0	0	4	2	0
8	0	0	0	2	0	0	0	2	4	2	4	0	0	2	0	0
9	0	0	4	0	0	0	0	0	4	2	0	2	0	2	0	2
a	0	0	0	2	0	0	0	2	4	2	0	0	0	2	4	0
b	0	0	0	0	0	0	4	0	4	2	0	2	0	2	0	2
c	0	0	0	0	2	2	2	2	0	2	0	2	2	0	2	0
d	0	0	0	2	2	2	2	0	0	2	0	0	2	0	2	2
e	0	0	2	2	2	2	0	0	0	2	2	0	2	0	0	2
f	0	0	2	0	2	2	0	2	0	2	2	2	2	0	0	0

(p_0,p_1)来表示概率:

$$(p_0,p_1)=\begin{cases}(0,0),\text{概率为}2^{0}\\(1,0),\text{概率为}2^{-2}\\(1,1),\text{概率为}2^{-3}\end{cases}\tag{4}$$

对于S盒的每一个可能的差分传播模式,均可用十维向量$(x_0,x_1,x_2,x_3,y_0,y_1,y_2,y_3,p_0,p_1)$来表示,含义为从输入差分$(x_0,x_1,x_2,x_3)$经过S盒传播到输出差分$(y_0,y_1,y_2,y_3)$的概率为$2^{-(2p_0+p_1)}$. 将S盒中所有可能的差分传播模式均用此方法表示,即可得到S盒的所有可能点的集合. 利用SageMath工具[19]中的inequality_generater方法,可生成包含有350个不等式的不等式集合,其刻画了所有可能差分传播模式,同时排除所有不可能差分传播模式.

由于该不等式集合中包含许多冗余不等式,且不等式数量过多时会影响到模型的求解效率,因此需要对不等式模型进行约简,保持模型准确的同时减少不等式的数量. 本文采用文献[6]中的MILP约简方法,其相比于文献[4]中的贪心算法效率更高,得到的不等式数量更少. 最终得到21个不等式来描述FBC算法S盒的差分传播. 不等式形式如下所示:

$$a_0x_0+\cdots+a_9x_9+a_{10}\geqslant 0\tag{5}$$

其中,$a_i\in\mathbb{Z},0\leqslant i\leqslant 10$.

(3)　额外条件

假设首轮的输入差分变量用$(x_0,x_1,\cdots,x_{127})$来表示,模型中根据需要针对输入差分添加额外的约束条件. 在不指定输入差分时,需要确保输入差分不全为0,即

$$\sum_{i=0}^{127}x_i\geqslant 1\tag{6}$$

假设要指定输入变量的差分值为$(v_0,v_1,\cdots,v_{127})$,则需添加以下128个约束条件:

$$\begin{cases}x_0=v_0\\x_1=v_1\\\quad\vdots\\x_{127}=v_{127}\end{cases}\tag{7}$$

(4)　目标函数

FBC算法中的差分特征概率主要由非线性部件S盒产生,模型中R轮差分特征概率可表示为

$$2^{-\sum_{i=0}^{R-1}\sum_{j=0}^{15}(2p_0^{16i+j}+p_1^{16i+j})}$$

因此目标函数设为

$$\min\sum_{i=0}^{R-1}\sum_{j=0}^{15}(2p_0^{16i+j}+p_1^{16i+j})\tag{8}$$

算法1中描述了基于比特的MILP差分特征搜索模型的建模方法,建模完成后可使用Gurobi求解器[20]进行求解.

算法1　基于比特的MILP差分特征搜索模型

输入:轮数r,输入差分值$d=(d_0,d_1,\cdots,d_{127})$

输出:r轮差分特征$\beta=(\beta_0,\beta_1,\cdots,\beta_r)$、概率DP

设定目标函数

设定初始输入差分约束条件

FOR $i=0$ TO $r-1$ DO:

　　设定异或及分支约束条件

　　设定S盒约束条件

　　设定线性层约束条件

设定Binary变量

求解模型,获取各个变量的值

返回r轮差分特征$\beta=(\beta_0,\beta_1,\cdots,\beta_r)$以及概率DP

3.2　分段统计法

根据差分特征的级联定义,如果一个r_1轮概率为p_1的差分特征的输出差分,与另一个r_2轮概率为p_2的差分特征的输入差分一致,则可以连接这两个差分特征获得一个(r_1+r_2)轮概率为$p_1\times p_2$的差分特征. 因此要提高差分特征的搜索效率,可以采用分段统计法,将R轮差分特征搜索模型分割为n个子模型,每个子模型搜索r_i轮的差分特征$(\sum_{i=0}^{n-1}r_i=R)$,通过差分特征的级联串联各个子模型的结果,最终得到长轮差分特征. 算法2中对分段统计法进行阐述.

以搜索FBC-128算法R轮的差分特征为例,对分段统计法进行详细介绍:

(1) 确定n个模型求解FBC-128差分特征的轮数

算法2 分段统计法

输入:n个MILP模型搜索差分特征的轮数$r_0, r_1, \cdots, r_{n-1}$

输出:R轮差分特征$\beta=(\beta_0, \beta_1, \cdots, \beta_R)$,概率DP

初始化β、DP

生成并计算r_0轮FBC-128差分特征模型

求解模型得到r_0轮差分特征$\Gamma_0=(\beta_0, \beta_1, \cdots, \beta_{r_0})$以及对应概率$\mathrm{DP}_0$

FOR $i=1$ TO $n-1$ DO:

设定输入差分为$\beta_m, m=\sum_{j=0}^{i-1} r_j$

生成并求解r_i轮FBC-128差分特征模型

求解模型得到r_i轮差分特征$\Gamma_i=(\beta_m, \beta_{m+1}, \cdots, \beta_{m+r_i})$, $m=\sum_{j=0}^{i-1} r_j$以及对应概率DP_i

更新β、DP

返回R轮差分特征β及其概率DP

$(r_0, r_1, \cdots, r_{n-1})$,且$\sum_{i=0}^{n-1} r_i = R$.

(2) 初始化β用于存放搜索到的R轮差分特征,初始化$DP=1$用于保存R轮差分特征的概率.

(3) 生成并求解首个MILP模型,根据模型的解获取r_0轮的差分特征$\Gamma_0=(\beta_0, \beta_1, \cdots, \beta_{r_0})$.

(4) 依次生成并求解第$i(1\leqslant i\leqslant n-1)$个MILP模型. 将第$i-1$个模型获得的差分特征中的输出差分$\beta_m$指定为第$i$个模型的输入差分,根据模型的解获取$r_i$轮的差分特征$\Gamma_i=(\beta_m, \beta_{m+1}, \cdots, \beta_{m+r_i})$及其概率$\mathrm{DP}_i$,其中$m=\sum_{j=0}^{i-1} r_j$. 将$\Gamma_i$写入数组$\beta$中,并更新$\mathrm{DP}=\mathrm{DP}\times\mathrm{DP}_i$.

(5) 在所有n个模型求解完毕后,输出R轮差分特征β及其概率DP. 若$\mathrm{DP}>2^{-128}$,则认为该差分特征β有效.

3.3 15轮FBC-128差分区分器

使用传统MILP方法搜索R轮差分特征时,通常会构建单个R轮差分特征搜索模型,求得最优的R轮差分特征. 然而在实际应用中,随着轮数的增加,模型中不等式规模会相应地扩大,受限于求解器有限的计算能力,可能无法在合理时间内得到有效解. 应用传统MILP搜索方法进行测试:经过近21 h的搜索,可得到FBC-128算法9轮概率为2^{-102}的差分特征;然而搜索10轮差分特征时,耗费近120 h所得到的差分特征概率最高到2^{-193},距离有效差分区分器(差分特征概率大于2^{-128})尚有较大差距. 因此在合理时间内,用传统MILP方法最多得到FBC-128算法的9轮差分区分器.

采用分段统计法寻找FBC-128算法的差分特征. 考虑到当前最长区分器为11轮,将总搜索轮数R设定为15轮,分段数定为三段,枚举所有的分段搜索方案并进行实际测试. 为保证搜索效率,设定前两个分段的轮数小于8,第三段的轮数小于7,此时的"三段式"分段方案共有21种,如表5所示. 在测试当前方案时,一旦发现区分器概率小于随机置换概率,则终止该方案并对当前差分区分器的轮数及其概率p进行记录.

搜索结果表明:"三分段"模式下的所有候选方案中,仅在6+5+4及5+6+4两种分段方案下可得到15轮差分区分器,概率为2^{-121}. 继续采用6+5+5及5+6+5两种方案寻找FBC-128算法的16轮差分区分器,其概率会小于随机置换概率,因此应用"分段统计法"得到的最长差分区分器为15轮. 经对比,二者仅在首轮输入差分值有所区别,在分段方案6+5+4下得到的15轮差分区分器可见表6,在5+6+4分段方案下,输入差分值为(00000000 00080000 04881202 00800208).

表5 FBC-128算法的分段统计法测试结果表

方案编号	分段方案	实际搜索轮数	$-\log_2 p$
1	2+7+6	2+7+3=12	2+59+54=115
2	3+6+6	3+6+1=10	4+77+30=111
3	3+7+5	3+7=10	4+109=113
4	4+5+6	4+5+3=12	12+50+58=120
5	4+6+5	4+6+3=13	12+59+54=125
6	4+7+4	4+7+1=13	12+77+27=116
7	5+4+6	5+4+4=13	18+35+61=114
8	5+5+5	5+5+3=13	18+48+53=119
9	**5+6+4**	**5+6+4=15**	**18+58+45=121**
10	5+7+3	5+7=12	18+85=103
11	6+3+6	6+3+4=13	26+27+64=117
12	6+4+5	6+4+3=13	26+42+58=126
13	**6+5+4**	**6+5+4=15**	**26+50+45=121**
14	6+6+3	6+6+1=13	26+67+31=124
15	6+7+2	6+7=13	26+76=102
16	7+2+6	7+2+4=13	32+30+64=126
17	7+3+5	7+3+2=12	32+47+42=121
18	7+4+4	7+4+1=12	32+60+28=120
19	7+5+3	7+5=12	32+83=115
20	7+6+2	—	—
21	7+7+1	—	—

4 16轮FBC-128算法密钥恢复攻击

针对FBC-128算法,应用表6中的15轮差分区分器$(\Delta_{\mathrm{in}}, \Delta_{\mathrm{out}})$向后扩展一轮发起16轮密钥恢复攻击,其中输入差分$\Delta_{\mathrm{in}}$=(00000000 00080000 848a1486 00800208),输出差分Δ_{out}=(812809a2 4c1f4b32 80080880 4c1f4b32). 密钥恢复步骤如下:

步骤1 确定要恢复的第16轮的子密钥. 在FBC第16轮加密过程中,共涉及到64比特子密钥. 因此,候选子密钥共有2^{64}个可能值,对应设立2^{64}个计数器$\delta_i, 0\leqslant i\leqslant 2^{64}$,并初始化清零.

表6　分段方案6+5+4得到的15轮FBC-128算法差分区分器

轮数	输入差分	概率
输入	00000000 00080000 848a1486 00800208	—
1	00080000 00002048 00880208 00002048	2^{-6}
2	00000000 00000000 00002048 00080000	2^{-14}
3	00000000 00000000 00080000 00000000	2^{-16}
4	00000000 00080000 00000000 00080000	2^{-16}
5	00080000 20480000 00000000 20480000	2^{-18}
6	00000000 02000080 20480000 02080080	2^{-26}
7	02000080 12020488 00080000 12020488	2^{-32}
8	00000000 88080000 12020488 8a080080	2^{-47}
9	88080000 a0480204 02000080 a0480204	2^{-55}
10	00000000 08080020 a0480204 80000020	2^{-72}
11	08080020 20488120 88080000 20488120	2^{-76}
12	00000000 80080880 20488120 880008a0	2^{-92}
13	80080880 c8204a24 08080020 c8204a24	2^{-98}
14	00000000 812809a2 c8204a24 01200122	2^{-115}
15	812809a2 4c1f4b32 80080880 4c1f4b32	2^{-121}

步骤2　均匀随机地选取2^{121}个明密文对(m,c)和(m',c'),其中$m\oplus m'=\Delta_{\text{in}}$,$c\oplus c'=\Delta_{\text{out}}$. 若$m$和$m'$为错误对,则舍弃. 比如根据FBC算法结构和其S盒的差分分布表,有如下结论:经过S盒后,$S(X_{15,0})$的值只能取2,8,9,b,d,f;$S(X_{15,1})$的值只能取0;$S(X_{15,2})$的值只能取3,4,7,b,c,f;$S(X_{15,3})$的值只能取0;$S(X_{15,4})$的值只能取2,3,6,7,a,b,e,f;$S(X_{15,5})$的值只能取0;$S(X_{15,6})$的值只能取1,3,5,7,b,f;$S(X_{15,7})$的值只能取3,7,8,9,d,e;$S(X_{15,8})$的值只能取0;$S(X_{15,25})$的值只能取3,7,8,9,d,e;$S(X_{15,26})$的值只能取1,3,5,7,b,f;$S(X_{15,27})$的值只能取3,4,7,b,c,f;$S(X_{15,28})$的值只能取2,3,4,5,9,a,c,f;$S(X_{15,29})$的值只能取4,5,6,7,9,b,c,e;$S(X_{15,30})$的值只能取1,2,6,a,d,e;$S(X_{15,31})$的值只能取2,3,6,7,a,b,e,f. 密文对差分值(c,c')通过逆线性变换即可得到第15轮的S盒组件输出差分,通过对比其差分值是否符合以上的条件可以筛选掉错误对.

在过滤后,剩余大概$2^{32.68}$个可能的输出差分值. 使用每一个可能的子密钥值进行第16轮的解密,得到第15轮的密文c_{11}和c'_{11},若其满足$c_{11}\oplus c'_{11}=\Delta_{\text{out}}$,则将对应计数器值加一.

步骤3　确定计数器T_δ的值,期望值为$N\times p$,若符合期望值则猜测目标部分子密钥的值为δ.

复杂度分析基于15轮区分器的16轮差分攻击数据复杂度为2^{121},时间复杂度为$2^{64}\times 2^{32.68}/16=2^{96.68}/16=2^{92.68}$.

5　结束语

本文基于MILP技术,针对FBC-128算法,提出了一种"分段统计法"的自动化搜索策略,提高了模型的搜索效率,给出了FBC-128算法抗差分密码分析的安全性评估. 结果表明:FBC-128算法存在概率为2^{-121}的15轮差分区分器. 进一步,我们将15轮差分区分器向后扩展1轮,对16轮FBC-128算法发起密钥恢复攻击,所需的数据复杂度为2^{121}个选择明文量,时间复杂度为$2^{92.68}$次16轮加密. 尽管攻击未能威胁到全轮FBC算法,但相比于已有结果,本文得到的差分区分器更长,且只需要更低的数据复杂度和时间复杂度就能够实现更高轮的密钥恢复攻击.

参考文献

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

[2] National Bureau of Standards. Data Encryption Standard: FIPS 46-3[S]. Washington: National Bureau of Standards, 1977.

[3] MOUHA N, WANG Q J, GU D W, et al. Differential and linear cryptanalysis using mixed-integer linear programming[C]//Information Security and Cryptology. Berlin: Springer, 2012(7537): 57-76.

[4] SUN S W, HU L, SONG L, et al. Automatic security evaluation of block ciphers with S-bP structures against related-key differential attack[C]//Information Security and Cryptology. Cham: Springer, 2014(8567): 39-51.

[5] SUN S W, HU L, WANG M Q, et al. Towards finding the best characteristics of some bit-oriented block ciphers and automatic enumeration of (related-key) differential and linear characteristics with predefined properties[R/OL]. (2015-02-09)[2023-02-16]. http://eprint.iacr.org/2014/747.

[6] SASAKI Y, TODO Y. New algorithm for modeling S-box in MILP based differential and division trail search[C]//Innovative Security Solutions for Information Technology and Communications. Cham: Springer, 2017(10543): 150-165.

[7] ZHU B Y, DONG X Y, YU H B. MILP-based differential attack on round-reduced GIFT[C]//Topics in Cryptology—CT-RSA 2019. Cham: Spring, 2019(11405): 372-390.

[8] BANIK S, PANDEY S K, PEYRIN T, et al. GIFT: A small present[C]//Cryptographic Hardware and Embedded Systems. Cham: Springer, 2017(10529): 321-345.

[9] BOURA C, COGGIA D. Efficient MILP modelings for sboxes and linear layers of SPN ciphers[J]. IACR Transactions on Symmetric Cryptology, 2020, 2020(3): 327-361.

[10] ZONG R, DONG X Y, CHEN H F, et al. Towards key-re-

covery-attack friendly distinguishers: Application to GIFT-128[J]. IACR Transactions on Symmetric Cryptology, 2021, 2021(1): 156-184.

[11] MAKARIM R H, ROHIT R. Towards tight differential bounds of Ascon: A hybrid usage of SMT and MILP[J]. IACR Transactions on Symmetric Cryptology, 2022, 2022(3): 303-340.

[12] SOOS M, NOHL K, CASTELLUCCIA C. Extending SAT solvers to cryptographic problems[C]//Theory and Applications of Satisfiability Testing—SAT 2009. Berlin: Springer, 2009(5584): 244-257.

[13] DOBRAUNIG C, EICHLSEDER M, MENDEL F, et al. Ascon v1.2: Lightweight authenticated encryption and hashing[J]. Journal of Cryptology, 2021, 34(3): 1-42.

[14] LI T, SUN Y. SuperBall: A new approach for MILP modelings of Boolean functions[J]. IACR Transactions on Symmetric Cryptology,2022, 2022(3): 341-367.

[15] 冯秀涛, 曾祥勇, 张凡, 等. 轻量级分组密码算法FBC[J]. 密码学报, 2019, 6(6): 768-785.
FENG X T, ZENG X Y, ZHANG F, et al. On the lightweight block cipher FBC[J]. Journal of Cryptologic Research, 2019, 6(6): 768-785. (in Chinese)

[16] REN B Q, CHEN J G, ZHOU S H, et al. Cryptanalysis of Raindrop and FBC[C]//Network and System Security. Cham: Springer, 2019(11928): 536-551.

[17] MATSUI M. On correlation between the order of S-boxes and the strength of DES[C]//EUROCRYPT 1994. Berlin: Springer, 1994(950): 366-375.

[18] ZHANG Y, LIU G Q, LI C, et al. Impossible differential cryptanalysis of FBC-128[J]. Journal of Information Security and Applications, 2022, 69(103279): 1-8.

[19] SAGEMATH. Sagemath 9.8[EB/OL]. (2023-02-11) [2023-02-16]. http://www.sagemath.org.

[20] GUROBI. Gurobi optimizer 10.0[EB/OL]. (2022-11-30) [2023-02-16]. http://www.gurobi.cn.

作者简介

赵 琪 男,1997年2月出生于山西省晋城市,现为桂林电子科技大学计算机与信息安全学院硕士研究生,研究方向为分组密码算法差分分析方法.
E-mail: wakeupzq@163.com

樊 婷 女,1993年10月出生于山西省忻州市,现为桂林电子科技大学计算机与信息安全学院博士研究生,研究方向为分组密码算法设计与分析.
E-mail: fanting0801@163.com

韦永壮 男,1976年12月出生于广西省百色市,现为桂林电子科技大学计算机与信息安全学院教授,博士生导师,主要研究方向为密码函数、对称密码算法设计与分析.
E-mail: walker_wyz@guet.edu.cn