LiCi分组密码算法的不可能差分分析

韦永壮[①②] 史佳利*[②③] 李灵琛[④⑤]

[①](桂林电子科技大学广西密码学与信息安全重点实验室 桂林 541004)

[②](桂林电子科技大学广西无线宽带通信与信号处理重点实验室 桂林 541004)

[③](桂林电子科技大学广西高校云计算与复杂系统重点实验室 桂林 541004)

[④](中国科学院大学 北京 100049)

[⑤](中国科学院软件研究所 北京 100190)

摘 要：LiCi是由Patil等人(2017)提出的轻量级分组密码算法。由于采用新型的设计理念，该算法具有结构紧凑、能耗低、占用芯片面积小等优点，特别适用于资源受限的环境。目前该算法的安全性备受关注，Patil等人声称：16轮简化算法足以抵抗经典的差分攻击及线性攻击。该文基于S盒的差分特征，结合中间相遇思想，构造了一个10轮的不可能差分区分器。基于此区分器，向前后各扩展3轮，并利用密钥编排方案，给出了LiCi的一个16轮的不可能差分分析方法。该攻击需要时间复杂度约为$2^{83.08}$次16轮加密，数据复杂度约为$2^{59.76}$选择明文，存储复杂度约为$2^{76.76}$数据块，这说明16轮简化的LiCi算法无法抵抗不可能差分攻击。

关键词：轻量分组密码算法；LiCi算法；不可能差分分析；差分特征

中图分类号：TP309 **文献标识码**：A **文章编号**：1009-5896(2019)07-1610-08

DOI: 10.11999/JEIT180729

Impossible Differential Cryptanalysis of LiCi Block Cipher

WEI Yongzhuang[①②] SHI Jiali[②③] LI Lingchen[④⑤]

[①](*Guangxi Key Laboratory of Cryptography and Information Security*, *Guilin University of Electronic Technology*, *Guilin* 541004 ,*China*)

[②](*Guangxi Key Laboratory of Wireless Wideband Communication and Signal Processing*, *Guilin University of Electronic Technology*, *Guilin* 541004 ,*China*)

[③](*Guangxi Colleges and University Key Laboratory of Cloud Computing and Complex Systems*, *Guilin University of Electronic Technology*, *Guilin* 541004 ,*China*)

[④](*University of Chinese Academy of Sciences*, *Beijing* 100049, *China*)

[⑤](*Institute of Software*, *Chinese Academy of Sciences*, *Beijing* 100190, *China*)

Abstract: LiCi algorithm is a newly lightweight block cipher. Due to its new design idea adopted by Patil *et al*, it has the advantages of compact design, low energy consumption and less chip area, thus is is especially suitable for resource-constrained environments. Currently, its security receives extensively attention, and Patil *et al*. claimed that the 16-round reduced LiCi can sufficiently resist both differential attack and linear attack. In this paper, a new 10-round impossible differential distinguisher is constructed based on the differential characteristics of the S-box and the meet-in-the-middle technique. Moreover, on the basis of this distinguisher, a 16-round impossible differential attack on LiCi is proposed by respectively extending 3-round forward and backward via the key scheduling scheme. This attack requires a time complexity of about $2^{83.08}$ 16-round

收稿日期：2018-07-19；改回日期：2018-10-29；网络出版：2019-03-18

*通信作者： 史佳利 jiali00@126.com

基金项目：国家自然科学基金(61572148, 61872103, 61561016), 广西研究生教育创新计划资助项目(YCBZ2018051), 获桂林电子科技大学研究生优秀学位论文培育项目(16YJPYSS12), 桂林电子科技大学研究生教育创新计划(2018YJCX45)

Foundation Items: The National Natural Science Foundation of China (61572148, 61872103, 61561016), The Innovation Project of Guangxi Graduate Education (YCBZ2018051), Guilin University of Electronic Technology Excellent Graduate Thesis Program (16YJPYSS12), The Innovation Project of Guilin University of Electronic Technology Graduate Education (2018YJCX45)

encryptions, a data complexity of about $2^{59.76}$ chosen plaintexts, and a memory complexity of $2^{76.76}$ data blocks, which illustrates that the 16-round LiCi cipher can not resist impossible differential attack.

Key words: Lightweight block cipher; LiCi cipher; Impossible differential cryptanalysis; Differential characteristic

1　引言

轻量级分组密码算法具有结构简单、速度快和便于软硬件实现等优点，广泛应用于物联网的微型设备中。对于RFID, WSN(无线传感器网络)、智能卡等环境受限的设备而言，保护敏感信息及重要数据的安全性至关重要，而轻量级加密算法通常围绕着低能耗、低成本的应用需求展开，所以其设计与分析备受关注[1]。目前，最常用的轻量级分组密码算法如LED[2], PRESENT[3], GIFT[4], Midori[5], LBlock[6], SMS4[7], SIMON[8], SPECK[8], CLEFIA[9]等，这些算法具有快速扩散能力、加解密相似、易于实现等优点。

最近，Patil等人[10]提出了一个轻量级分组密码算法——LiCi。该算法基于平衡Feistel结构，并采用轻量级S盒及简单移位操作等新型设计理念，即在比较少的轮运算下容易产生最大数目的活跃S盒。该算法具有结构紧凑、能耗低、占用面积小等特性，比如硬件实现仅需要花销大约1153个等价门，非常适用于资源受限的环境。特别地，Patil等人声称：16轮的简化算法足以抵抗经典的差分攻击及线性攻击。然而该算法是否能抵抗不可能差分分析仍有待进一步研究。

不可能差分分析最初是由Knudsen[11]和Biham等人[12]分别独立提出，也是目前最有效的密码分析方法之一[13–18]。它的攻击原理主要是利用概率为0的不可能差分特性，过滤错误密钥，直至恢复出正确的密钥。本文基于LiCi算法本身的结构特性，结合S盒特有的差分性质，对该算法进行不可能差分分析：通过利用一个4轮加密方向的差分及6轮解密方向的差分，结合中间相遇思想，构造了一个10轮的不可能差分区分器；由此向前后各扩展3轮，从而形成对16轮的LiCi算法的不可能差分分析。在密钥恢复阶段，采用分步猜测子密钥的方法，结合S盒输入输出差分特征，以降低攻击所需的时间复杂度。结果证实了16轮简化LiCi算法并不安全。

本文后续部分安排如下：第2节对LiCi算法的结构进行介绍；第3节介绍LiCi算法的一些特性；第4节给出了LiCi算法的10轮不可能差分区分器的构造及16轮不可能差分分析的过程；第5节对全文进行总结。

2　LiCi算法介绍

LiCi算法[10]的分组长度为64 bit，密钥长度128 bit，其迭代轮数为31轮。轮函数如图1所示，包括字节替换、密钥加、循环移位、异或这4种运算，其中字节替换是唯一的非线性函数，采用8个并列的S盒。

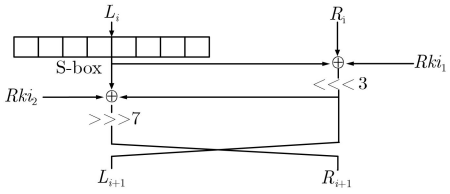

图 1 LiCi算法的轮函数

设第i轮的输入为X_i，输出为X_{i+1}, LiCi算法的左分支和右分支分别记为L_i, R_i $(0 \le i < 31)$，则状态(L_i, R_i)至状态(L_{i+1}, R_{i+1})的更新过程为

$$\begin{aligned} &X_i = L_i || R_i, S_i = S[L_i], \\ &L_{i+1} = [S_i \oplus R_i \oplus \mathrm{Rk}i_1] <<< 3, \\ &R_{i+1} = [S_i \oplus L_{i+1} \oplus \mathrm{Rk}i_2] >>> 7 \end{aligned} \tag{1}$$

密钥编排　密钥长度为128 bit，为$K = k_{127}k_{126}k_{125}\cdots k_2k_1k_0$。每轮取最右端的64 bit作为轮密钥，记为$\mathrm{Rk}i = (\mathrm{Rk}i_2, \mathrm{Rk}i_1)$，其中，$\mathrm{Rk}i_2$应用于左分支，而$\mathrm{Rk}i_1$应用于右分支，长度均为32 bit。密钥更新过程为：

(1) $K = k_{127}k_{126}k_{125}\cdots k_2k_1k_0$, $\mathrm{Rk0}_1[31-0] = K[31-0]$, $\mathrm{Rk0}_2[31-0] = K[63-32]$;

(2) For $i \leftarrow 0$ to 30：

$K <<< 13$;

$K[3-0] \leftarrow S(K[3-0])$, $K[7-4] \leftarrow S(K[7-4])$, $K[63-59] \leftarrow K[63-59] \oplus \mathrm{RC}^i$;

$\mathrm{Rk}i_1[31-0] = K[31-0]$,

$\mathrm{Rk}i_2[31-0] = K[63-32]$

3　LiCi的算法性质

3.1　S盒差分性质

本节采用类似于文献[19]的方法，针对LiCi算法的S盒的差分分布表(DDT)进行分析，对S盒输入输出的差分特征进行总结。

性质1 当S盒的输入/输出差分为某些定值时，输出/输入差分存在一些对应规律，如表1所示。其中“*”表示差分未知，“0”表示差分为0，“1”表示差分为1。S盒的这些差分特征应用于算法的加/解密推导，可以有效地拓展不可能差分区分器的长度。

性质2 若输入(输出)差分为某些值/特定值时，其对应的输出(输入)差分为特定值/某些值存在对应的概率。LiCi算法S盒具体输入输出的差分特征对应的概率详见表2。

表 1 S盒输入/输出差分特征

输入差分	0001	0100	1000	1100	1101	**11	**1*	**0*	***1	***0
输出差分	1***	***1	***1	1**0	0***	0001	0010	0011	0100	0101

表 2 S盒输入/输出差分特征概率

输入差分	**11	**1*	***1	****	0001	0***	*0**	01**	*000
输出差分	0001	0010	0100	1000	1***	****	****	****	****
概率	1/4	1/8	1/8	1/16	1/8	1/2	1/2	1/4	1/8

注记1(S盒差分概率) 给定输入(输出)差分经过S盒(逆运算)后，其对应的输出(输入)差分存在对应的概率。例如：若输出差分为0001，则输入差分为$**11$的概率为$1/4$。也就是说输出差分为0001时，经S盒逆运算后，其输入差分可能为0011, 0111, 1011, 1111这4种情况，可表示为$**11$，其对应的概率为：P(0011)=P(0111)=P(1011)=P(1111)=4/16=1/4，则该结论成立(同理可证实性质1及性质2的其它结论)。在进行密钥恢复时，若考虑到S盒特殊输入输出差分中存在的概率，可有效地降低密钥恢复的复杂度，有助于数据集中的明密文对筛选。

3.2 密钥编排性质

对于LiCi算法，设初始密钥为$K=k_{127}k_{126}k_{125}\cdots k_2k_1k_0$。若第$i$轮子密钥记为Rki，而其左分支及右分支的子密钥分别记为Rki_2, Rki_1。

性质3 对于LiCi算法，第1～3轮及第14～16轮的轮子密钥中存在一些等价关系(若$a=b$或$a=b\oplus 1$，这里将a与b之间的关系视为等价)。

证明 设初始密钥为$K_0=k_{127}k_{126}k_{125}\cdots k_2k_1k_0$，第1～3轮的轮密钥(分别记为Rk$i_j$, $0\le i<3, j\in(1,2)$)中有38 bit存在等价关系。具体的密钥比特为

$$\begin{aligned}&k_{37},k_{36},\cdots,k_0\\&=\mathrm{Rk0}_2[5,4,\cdots,0]\cup\mathrm{Rk0}_1[31,30,\cdots,0]\\&=\mathrm{Rk1}_2[18,17,\cdots,0]\cup\mathrm{Rk1}_1[31,30,\cdots,13]\\&=\mathrm{Rk2}_2[31,30,\cdots,0]\cup\mathrm{Rk2}_1[31,30,\cdots,26]\end{aligned}\quad(2)$$

其中，$\mathrm{Rk2}_2[27]=k_{33}\oplus 1$)。

同理，设第14轮的轮密钥为$K_{13}=k'_{127}k'_{126}k'_{125}\cdots k'_2k'_1k'_0$，第14～16轮的轮密钥(分别记为Rk$i_j, 13\le i<16, j\in(1,2)$)有38 bit存在等价关系，具体的密钥比特为

$$\begin{aligned}&k'_{37},k'_{36},\cdots,k'_0\\&=\mathrm{Rk13}_2[5,4,\cdots,0]\cup\mathrm{Rk13}_1[31,30,\cdots,0]\\&=\mathrm{Rk14}_2[18,17,\cdots,0]\cup\mathrm{Rk14}_1[31,30,\cdots,13]\\&=\mathrm{Rk15}_2[31,30,\cdots,0]\cup\mathrm{Rk15}_1[31,30,\cdots,26]\end{aligned}\quad(3)$$

其中，$\mathrm{Rk15}_2[30]=k'_{36}\oplus 1$，$\mathrm{Rk15}_2[29]=k'_{35}\oplus 1$，$\mathrm{Rk15}_2[28]=k'_{34}\oplus 1$。 证毕

注记2 经过分析算法的密钥编排，可以得到连续3轮轮子密钥之间的关系。在密钥恢复过程中，该性质可以极大地缩小需要猜测的密钥空间，进而降低攻击的时间复杂度。

4 16轮LiCi算法不可能差分分析

4.1 10轮不可能差分区分器

本节基于LiCi算法自身的结构特性，结合性质1及中间相遇思想，推导出一个10轮不可能差分区分器(如图2所示)。即将概率为1的4轮加密差分方向及6轮解密差分方向连接起来，进而构成一个10轮不可能差分区分器。

定理1 设明文与密文分别表示为P_i和C_i，其中i表示轮数。若第4轮的输入差分满足$\Delta P_3=(\Delta L_3,\Delta R_3)$，其中$\Delta L_3=0$(这里“0”表示32 bit的差分均为0)，$\Delta R_3=$[0000, 0000, 0000, 0000, 0001, 0000, 0000, 0000]，则第13轮的输出差分不可能满足$\Delta C_{13}=(\Delta L_{13},\Delta R_{13})$，其中$\Delta L_{13}=$[0000, 0000, 0001, 0000, 0000, 0000, 0000, 0000]，$\Delta R_3=$[0000, 0000, 0000, 0000, 0010, 0000, 0000, 0000]。

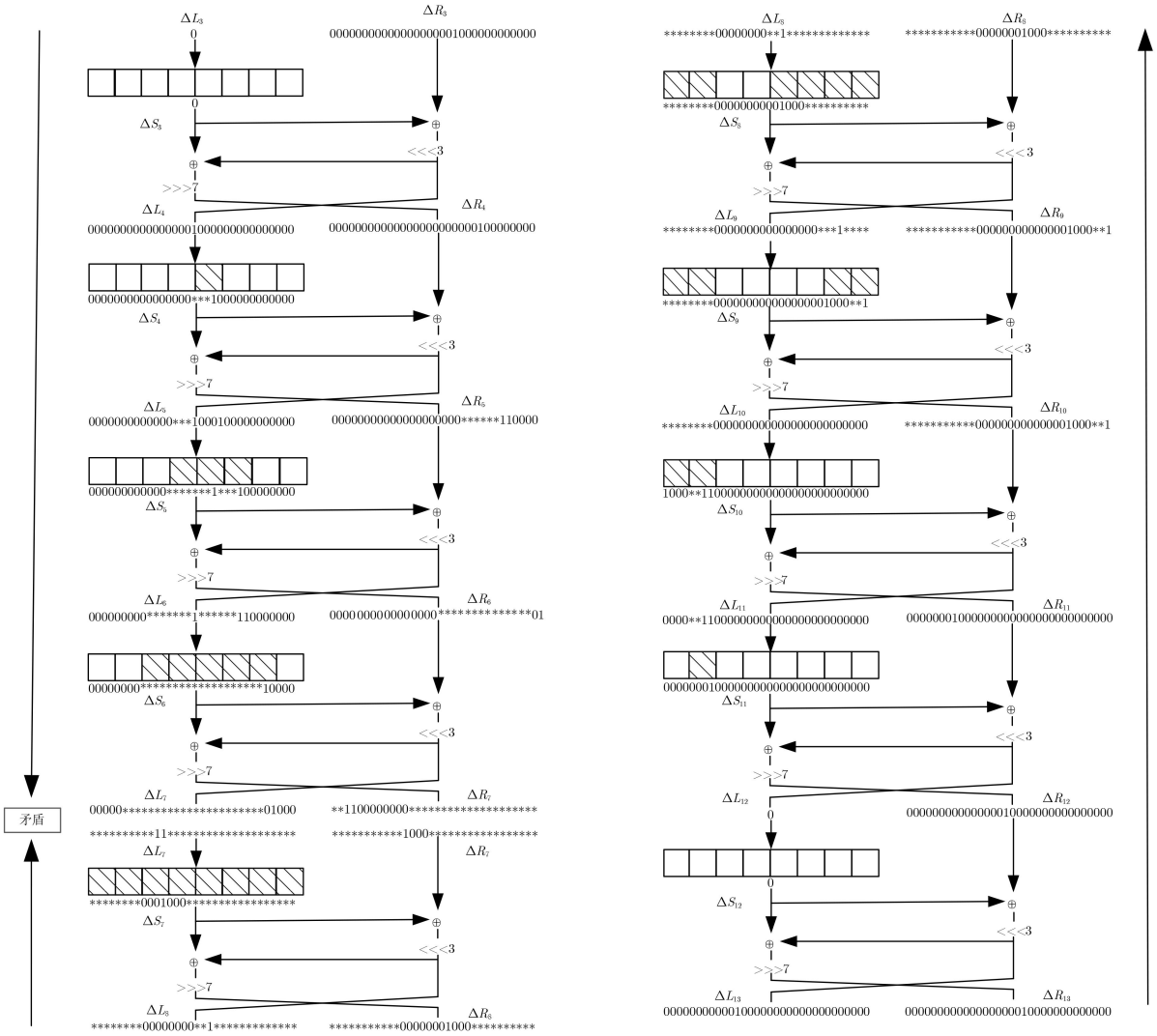

图 2 LiCi算法的10轮不可能差分路线

证明　加密方向，设第4轮输入差分$\Delta P_3=(\Delta L_3,\Delta R_3)$，其中左分支差分满足$\Delta L_3=0$(“0”表示32 bit的差分均为0)，右分支差分满足$\Delta R_3=[0000,0000,0000,0000,0001,0000,0000,0000]$。此时，左分支经过非线性替换(S盒)后的输出差分仍为0，随后，经过右分支异或及右循环3位操作后的输出差分，即第5轮左分支的输入差分满足$\Delta L_4=[0000,0000,0000,0000,1000,0000,0000,0000]$(移位变换不影响差分)，类似第5轮右分支的输入差分应满足$\Delta R_4=[0000,0000,0000,0000,0000,0001,0000,0000]$。推导过程中结合LiCi算法的性质及其S盒的差分特征，经过4轮迭代后，得到第7轮左、右分支输出差分分别满足$\Delta L_7=[0000,0***,**11,****,****,****,***0,1000]$，$\Delta R_7=[**11,0000,0000,****,****,****,****,****]$，可知右分支第16 bit的输出差分为0。

解密方向，设第13轮左分支输出差分满足$\Delta L_{13}=[0000,0000,0001,0000,0000,0000,0000,0000]$，右分支输出差分满足$\Delta R_3=[0000,0000,0000,0000,0010,0000,0000,0000]$。同样地，当进行非线性替换运算(S盒)时，若符合推导的差分特征，则根据其特征进行推演(由性质1得到)，获得第8轮左、右分支的输入差分分别满足$\Delta L_7=[****,****,**11,****,****,****,****,****]$，$\Delta R_7=[****,****,***1,000*,****,****,****,****]$，可知右分支第16 bit的输入差分为1，通过加解密路径输入输出差分的对比，这与加密方向的结论相矛盾。　　证毕

具体差分传播过程，请参见图2。

4.2 LiCi算法的16轮不可能差分分析

在10轮不可能差分区分器的基础上，前后各扩展3轮，对16轮的LiCi算法进行不可能差分分析(如图3所示)。具体的分析主要分为两个阶段：数据收

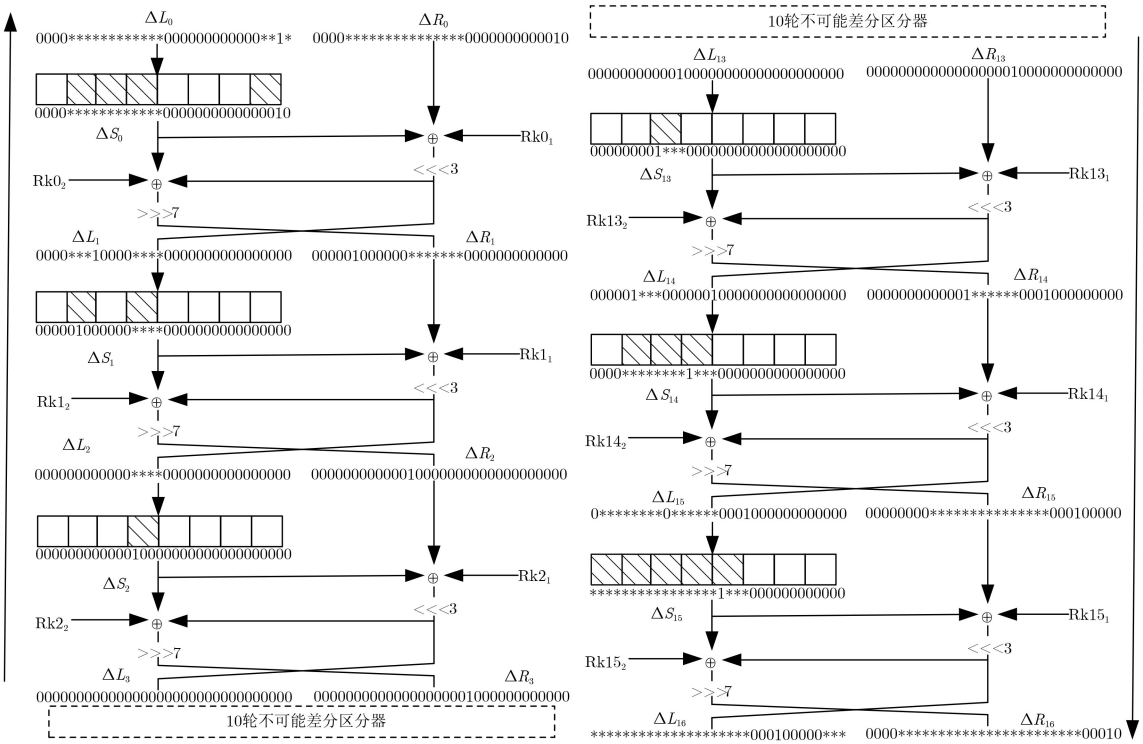

图 3 LiCi算法的16轮不可能差分分析

集、密钥恢复过程。在密钥恢复的过程中，利用分步猜测轮密钥的方法，对其进行密钥恢复。具体攻击步骤如下：

数据收集 首先选取2^m个数据集，使得每个数据集中的明文对(P, P')的差分满足$\Delta P=(\Delta L_0, \Delta R_0)$(见图3)，则共有34 bit可取任意值，即每个数据集中大约可形成2^{67}个明文对。另一方面，第16轮的输出密文的差分需满足$\Delta C=(\Delta L_{16}, \Delta R_{16})$，共有18 bit取固定值，则可以筛选出$2^{m+49}$个明密文对。

密钥恢复步骤：

(1) 对于选取的每一个明密文对，筛选出那些差分满足ΔS_0的明密文对，其中$\Delta S_0=S_0\oplus S'_0$，$S_0=S(L_0)$, $S'_0=S(L'_0)$。若结果满足该差分，则此数据对留下，反之，则将其删除。由于$\Delta S_0[3\sim 0]=0010$，且$S_0(**1*)=0010$的概率为1/8，筛选后大约剩余2^{m+46}明密文对。接着筛选出差分满足$\Delta L_1[30\sim 28, 24\sim 20]=[0,0,0,1,0,0,0,0]$, $\Delta R_1[20, 12\sim 9]=[0,0,0,0,0]$的明密文对，其中$\Delta L_1=L_1\oplus L'_1=(\Delta S_0\oplus \Delta R_0)$，$\Delta R_1=R_1\oplus R'_1=(\Delta S_0\oplus \Delta L_0)$，筛选后大约剩余$2^{m+33}$明密文对。解密一轮，筛选出那些差分满足$\Delta S_{15}[15, 2\sim 0]=[1,0,0,0]$, $\Delta R_{15}[31\sim 24]=[0,0,0,0,0,0,0,0]$的明密文对，其中，$\Delta S_{15}=(\Delta R_{16})_{<<<7}\oplus \Delta L_{16}$，$\Delta R_{15}=(\Delta L_{16})_{>>>3}\oplus \Delta S(L_{15})$，所以筛选后大约剩余$2^{m+21}$明密文对。

(2) 猜测第1轮右分支$\mathrm{Rk0}_1[24\sim 21]$这4 bit密钥的值。对于剩余的每一个明密文对，筛选出差分满足$\Delta S_1[27\sim 24]=0100$的数据对，其中，$\Delta S_1=S_1\oplus S'_1$，$S_1=S(L_1)$，$L_1=[S_0\oplus R_0\oplus \mathrm{Rk0}_1]_{<<<3}$。由于$S_1(***1)=0100$的概率为1/8，所以筛选后大约剩余$2^{m+18}$明密文对。

(3) 猜测第16轮左分支$\mathrm{Rk15}_2[15\sim 12]$这4 bit密钥的值。对于剩余的每个明密文对，筛选出差分满足$\Delta L_{15}[15\sim 12]=0001$的明密文对，其中，$\Delta L_{15}=L_{15}\oplus L'_{15}$，$L_{15}=S_{15}^{-1}=S^{-1}(S_{15})$，$S_{15}=S(L_{15})=(R_{16})_{<<<7}\oplus L_{16}\oplus \mathrm{Rk15}_2$，由于$\Delta S_{15}^{-1}(1***)=0001$的概率约为1/8，所以筛选后大约剩余$2^{m+15}$明密文对。

(4) 猜测第16轮左分支$\mathrm{Rk15}_2[31\sim 28]$这4 bit密钥的值。对于剩余的每个明密文对，筛选出满足$\Delta L_{15}[31\sim 28]=0***$的数据对，其中，$\Delta L_{15}=L_{15}\oplus L'_{15}$，$L_{15}=S_{15}^{-1}=S^{-1}(S_{15})$，又由于$\Delta S_{15}^{-1}(****)=0***$的概率约为1/2，所以筛选后大约剩余$2^{m+14}$明密文对。

(5) 猜测第16轮左分支$\mathrm{Rk15}_2[23\sim 20]$这4 bit密钥的值。对于剩余的明密文对，筛选差分满足$\Delta L_{15}[23\sim 20]=*0**$的明密文对，其中，$\Delta L_{15}=L_{15}\oplus L'_{15}$，$L_{15}=S_{15}^{-1}=S^{-1}(S_{15})$，$S_{15}=S(L_{15})=(R_{16})_{<<<7}\oplus L_{16}\oplus \mathrm{Rk15}_2$，又由于$\Delta S_{15}^{-1}(****)=*0**$的概率约为1/2，所以筛选后大约剩余$2^{m+13}$明密文对。

(6) 猜测第16轮右分支密钥$\mathrm{Rk15}_1[16\sim13]$和第15轮左分支密钥$\mathrm{Rk14}_2[23\sim20]$这8 bit密钥的值。对于剩余的每一个明密文对，筛选出差分满足$\Delta L_{14}[23\sim20]=*000$的明密文对，其中，$\Delta L_{14}=L_{14}\oplus L'_{14}$，$L_{14}=S_{14}^{-1}=S^{-1}(S_{14})$，$S_{14}=S(L_{14})=(R_{15})_{<<<7}\oplus L_{15}\oplus \mathrm{Rk14}_2$，由于$\Delta S_{14}^{-1}(****)=*000$的概率约为1/8，所以筛选后大约剩余$2^{m+10}$明密文对。

(7) 猜测第16轮左分支密钥$\mathrm{Rk15}_2[19\sim16,11\sim9]$、右分支密钥$\mathrm{Rk15}_1[12\sim9]$和第15轮左分支密钥$\mathrm{Rk14}_2[19\sim16]$这15 bit密钥的值。其中，第15轮左分支子密钥满足$\mathrm{Rk14}_2[18\sim16]=\mathrm{Rk15}_2[31\sim29]=k'_{37}\sim k'_{35}$(由性质3得到)，因此只需猜测12 bit子密钥。对于剩余的明密文对，筛选出差分满足$\Delta L_{14}[19\sim16]=0001$的明密文对，其中$\Delta L_{14}=L_{14}\oplus L'_{14}$，$L_{14}=S_{14}^{-1}=S^{-1}(S_{14})$，$S_{14}=S(L_{14})=(R_{15})_{<<<7}\oplus L_{15}\oplus \mathrm{Rk14}_2$，由于$\Delta S_{14}^{-1}(1***)=0001$的概率为1/8，所以筛选后大约剩余$2^{m+7}$明密文对。

(8) 猜测第16轮左分支密钥$\mathrm{Rk15}_2[27\sim24]$、右分支密钥$\mathrm{Rk15}_1[20\sim17]$和第15轮左分支密钥$\mathrm{Rk14}_2[27\sim24]$这12 bit密钥的值。对于剩余的明密文对，筛选出差分满足$\Delta L_{14}[27\sim24]=01**$的明密文对，其中，$\Delta L_{14}=L_{14}\oplus L'_{14}$, $L_{14}=S_{14}^{-1}=S^{-1}(S_{14}), S_{14}=S(L_{14})=(R_{15})_{<<<7}\oplus L_{15}\oplus \mathrm{Rk14}_2$,由于$\Delta S_{14}^{-1}(****)=01**$的概率约为1/4，所以筛选后大约剩余$2^{m+5}$明密文对。

(9) 猜测第16轮左分支密钥$\mathrm{Rk15}_2[8\sim6]$、右分支密钥$\mathrm{Rk15}_1[8\sim6]$和第15轮左分支密钥$\mathrm{Rk14}_2[15\sim13]$、右分支密钥$\mathrm{Rk14}_1[16\sim13]$及第14轮左分支密钥$\mathrm{Rk13}_2[23\sim20]$这17 bit密钥的值。其中，第15轮左分支子密钥应满足$\mathrm{Rk14}_2[15\sim13]=\mathrm{Rk15}_2[28\sim26]=k'_{34}\sim k'_{32}$，右分支子密钥应满足$\mathrm{Rk14}_1[16\sim13]=\mathrm{Rk15}_2[10\sim7]=k'_{16}\sim k'_{13}$(由性质3得到)，所以只需猜测10 bit密钥。接下来，对于剩余的每一个明密文对，需要筛选出差分满足$\Delta L_{13}[23\sim20]=0001$的明密文对，其中，$\Delta L_{13}=L_{13}\oplus L'_{13}$，$L_{13}=S_{13}^{-1}=S^{-1}(S_{13})$，$S_{13}=S(L_{13})=(R_{14})_{<<<7}\oplus \mathrm{Rk13}_2$，由于$\Delta S_{13}^{-1}(1***)=0001$的概率为1/8，所以筛选后大约剩余$2^{m+2}$明密文对；解密一轮，需要筛选出差分满足$\Delta S_{14}[30\sim28,19]=[0,0,0,1]$，$\Delta R_{14}[27\sim20]=[0,0,0,0,0,0,0,0]$的明密文对，其中$\Delta S_{14}=(\Delta R_{15})_{<<<7}\oplus\Delta L_{15}$, $\Delta R_{14}=(\Delta L_{15})_{>>>3}\oplus\Delta S(L_{14})$，此时大约剩余$2^{m+10}$明密文对。最后筛选出差分满足$\Delta S_{13}[25\sim23]=[0,0,1]$，$\Delta R_{13}[22\sim20]=[0,0,0]$的明密文对，其中，$\Delta S_{13}=(\Delta R_{14})_{<<<7}\oplus\Delta L_{14}$, $\Delta R_{13}=(\Delta L_{14})_{>>>3}\oplus\Delta S(L_{13})$，所以筛选后大约剩余$2^{m-16}$明密文对。

(10) 猜测第1轮右分支密钥$\mathrm{Rk0}_1[20\sim9]$、左分支密钥$\mathrm{Rk0}_2[23\sim20]$和第2轮右分支密钥$\mathrm{Rk1}_1[16\sim13]$这20 bit密钥的值。针对剩余的明密文对，筛选出差分满足$\Delta S_2[19\sim16]=1000$的明密文对，其中，$\Delta S_2=S(L_2)\oplus S(L'_2)$，由于$\Delta S_2(****)=1000$的概率约为1/16，经过筛选，大约剩余$2^{m-20}$明密文对。接着筛选出那些差分满足$\Delta L_2[22\sim20]=[0,0,0]$，$\Delta R_2[12\sim9]=[0,0,0,0]$的明密文对，其中，$\Delta L_2=L_2\oplus L'_2=(\Delta S_1\oplus\Delta R_1)$, $\Delta R_2=R_2\oplus R'_2=(\Delta S_1\oplus\Delta L_1)$，最后大约剩余$2^{m-27}$明密文对。

复杂度分析　密钥恢复的过程中共猜测了78 bit子密钥，并利用选取的所有明密文对，排除所有的错误密钥，筛选出唯一的正确密钥。注意到经过以上筛选后，大约剩余$2^{78}\times\left(1-2^{-4}\right)^{2^{m-16}}$个候选密钥。当剩余的候选密钥数目小于等于1时，才能保证恢复出唯一的正确密钥，即$2^{78}\times\left(1-2^{-4}\right)^{2^{m-16}}\le 1$时，解得$m\approx25.76$。此时错误率为：$P=\left(1-2^{-4}\right)^{2^{m-16}}=\mathrm{e}^{-2^{m-12}}=\mathrm{e}^{-2^{13.76}}=2^{-2^{14.29}}$。因此，对16轮的LiCi算法进行不可能差分分析的数据复杂度和存储复杂度分别为：$2^{m+34}=2^{59.76}$选择明文和$2^{m+49}\times4=2^{76.76}$数据块。由性质3分析可知，所猜测的78 bit子密钥实际可以获得54 bit主密钥，因而还需遍历剩余的74 bit主密钥，即可恢复出所有的密钥。又因为数据收集的时间复杂度为$2^{m+34}=2^{59.76}$，由表3可知该攻击总的时间复杂度为$2^{83.08}$次16轮加密。

从表4可以看出：16轮的LiCi算法足以抵抗差分分析及线性分析，但无法抵抗不可能差分分析。

5　结论

文献[10]声称16轮简化LiCi算法足以抵抗差分分析、线性分析等分析方法，并将16轮视为安全界限。本文对LiCi算法进行不可能差分分析：利用算法结构特性及S盒差分特征，构造了一个10轮的不可能差分区分器，并向前后分别扩展3轮，从而构建了16轮LiCi算法的不可能差分分析。整个攻击的数据复杂度为$2^{59.76}$选择明文，时间复杂度为$2^{83.08}$次16轮加密，存储复杂度为$2^{76.76}$数据块。分析结果表明：16轮的LiCi算法无法抵抗不可能差分攻击。

表 3 不可能差分分析16轮LiCi算法的密钥恢复各步骤的时间复杂度

步骤	恢复的密钥比特	猜测比特数目	时间复杂度(单位：次16轮加密所用的时间)
(1)	–	0	$2^{m+49}\times\frac{1}{8\times16}=2^{m+42}$
(2)	$\mathrm{Rk0}_1[24\sim21]$	4	$2^{m+21}\times2^4\times\frac{1}{8\times16}=2^{m+18}$
(3)	$\mathrm{Rk15}_2[15\sim12]$	4	$2^{m+18}\times2^4\times2^4\times\frac{1}{8\times16}=2^{m+19}$
(4)	$\mathrm{Rk15}_2[31\sim28]$	4	$2^{m+15}\times2^4\times2^4\times2^4\times\frac{1}{8\times16}=2^{m+20}$
(5)	$\mathrm{Rk15}_2[23\sim20]$	4	$2^{m+14}\times2^4\times2^4\times2^4\times2^4\times\frac{1}{8\times16}=2^{m+23}$
(6)	$\mathrm{Rk15}_1[16\sim13]$, $\mathrm{Rk14}_2[23\sim20]$	8	$2^{m+13}\times2^4\times2^4\times2^4\times2^4\times2^8\times\frac{1}{8\times16}=2^{m+30}$
(7)	$\mathrm{Rk15}_2[19\sim16,11\sim9]$, $\mathrm{Rk15}_1[12\sim9]$, $\mathrm{Rk14}_2[19\sim16]$	12	$2^{m+10}\times2^{16}\times2^8\times2^{12}\times2\times\frac{1}{8\times16}=2^{m+40}$
(8)	$\mathrm{Rk15}_2[27\sim24]$, $\mathrm{Rk15}_1[20\sim17]$, $\mathrm{Rk14}_2[27\sim24]$	12	$2^{m+7}\times2^{24}\times2^{12}\times2^{12}\times2\times\frac{1}{8\times16}=2^{m+49}$
(9)	$\mathrm{Rk15}_2[8\sim6]$, $\mathrm{Rk15}_1[8\sim6]$, $\mathrm{Rk15}_2[15\sim13]$, $\mathrm{Rk14}_1[16\sim13]$	10	$2^{m+5}\times2^{24}\times2^{12}\times2^{12}\times2^{10}\times2\times\frac{1}{8\times16}=2^{m+56}$
(10)	$\mathrm{Rk0}_1[20\sim9]$, $\mathrm{Rk0}_2[23\sim20]$, $\mathrm{Rk1}_1[16\sim13]$	20	$2^{m-16}\times2^{58}\times2^{20}\times3\times\frac{1}{8\times16}+2^{74}=2^{m+56.58}+2^{74}$

表 4 LiCi算法的攻击结果对比

分析方法	攻击轮数	选择明文量	时间复杂度	存储复杂度	参考文献
线性分析	16	2^{106}	–	–	文献[10]
差分分析	16	2^{96}	–	–	文献[10]
不可能差分分析	16	$2^{59.76}$	$2^{83.08}$	$2^{76.76}$	本文

参 考 文 献

[1] BIRYUKOV A and PERRIN L. State of the art in lightweight symmetric cryptography[R]. Cryptology ePrint Archive: Report 2017/511, 2017: 1–11.

[2] GUO Jian, PEYRIN T, POSCHMANN A, *et al*. The LED block cipher[C]. Proceedings of the 13th International Workshop on Cryptographic Hardware and Embedded Systems, Nara, Japan, 2011: 326–341.

[3] BOGDANOV A, KNUDSEN L R, LEANDER G, *et al*. PRESENT: An ultra-lightweight block cipher[C]. Proceedings of 9th International Workshop on Cryptographic Hardware and Embedded Systems, Vienna, Austria, 2007: 450–466.

[4] BANIK S, PANDEY S K, PEYRIN T, *et al*. GIFT: A small present[C]. Proceedings of the 19th International Conference on Cryptographic Hardware and Embedded Systems, Taipei, China, 2017: 321–345.

[5] BANIK S, BOGDANOV A, ISOBE T, *et al*. Midori: A block cipher for low energy[C]. Proceedings of the 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, 2015: 411–436.

[6] WU Wenling and ZHANG Lei. LBlock: A lightweight block cipher[C]. Proceedings of the 9th International Conference on Applied Cryptography and Network Security, Nerja, Spain, 2011: 327–344.

[7] 国家商用密码管理办公室. 无线局域网产品使用的SMS4密码算法[EB/OL]. http://www.oscca.gov.cn/sca/c100061/201611/1002423/files/330480f731f64e1ea75138211ea0dc27.pdf, 2018.
Office of State Commercial Cipher Administration. Block cipher for WLAN products-SMS4[EB/OL]. http://www.oscca.gov.cn/sca/c100061/201611/1002423/files/330480f731f64e1ea75138211ea0dc27.pdf, 2018.

[8] BEAULIEU R, TREATMAN-CLARK S, SHORS D, *et al*. The SIMON and SPECK lightweight block ciphers[C]. Proceedings of the 52nd ACM/EDAC/IEEE Design Automation Conference, San Francisco, USA, 2015: 1–6.

[9] AOKI K, ICHIKAWA T, KANDA M, *et al*. Camellia: A 128-bit block cipher suitable for multiple platforms-design and analysis[C]. Proceedings of the 7th Annual International Workshop on Selected Areas in Cryptography, Ontario, Canada, 2000: 39–56.

[10] PATIL J, BANSOD G, and KANT K S. LiCi: A new ultra-lightweight block cipher[C]. Proceedings of 2017 International Conference on Emerging Trends & Innovation in ICT, Pune, India, 2017: 40–45.

[11] KNUDSEN L R. DEAL-a 128-bit block cipher[R]. Technical Report, 1998.

[12] BIHAM E, BIRYUKOV A, and SHAMIR A. Cryptanalysis

of Skipjack Reduced to 31 Rounds Using Impossible Differentials[M]. Berlin, Germany, 1999: 12–23.

[13] BIHAM E and SHAMIR A. Differential cryptanalysis of DES-like cryptosystems[C]. Proceedings of the 10th Annual International Cryptology Conference on Advances in Cryptology, 1991: 2–21.

[14] TJUAWINATA I, HUANG Tao, and WU Hongjun. Improved differential cryptanalysis on generalized feistel schemes[C]. Proceedings of the 18th International Conference on Cryptology in India, Chennai, India, 2017: 302–324.

[15] LIU Ya, GU Dawu, LIU Zhiqiang, *et al*. Impossible differential attacks on reduced-round LBlock[C]. Proceedings of the 8th International Conference on Information Security Practice and Experience, Hangzhou, China, 2012: 97–108.

[16] KONDO K, SASAKI Y, TODO Y, *et al*. Analyzing key schedule of SIMON: Iterative key differences and application to related-key impossible differentials[C]. Proceedings of the 12th International Workshop on Security, Hiroshima, Japan, 2017: 141–158.

[17] MEHRDAD A, MOAZAMI F, and SOLEIMANY H. Impossible differential cryptanalysis on Deoxys-BC-256[R]. Cryptology ePrint Archive: Report 2018/048, 2018.

[18] SHAHMIRZADI A R, AZIMI S A, SALMASIZADEH M, *et al*. Impossible differential cryptanalysis of reduced-round Midori64 block cipher[J]. *ISeCure*, 2018, 10(1): 3–14.

[19] TEZCAN C. Improbable differential attacks on Present using undisturbed bits[J]. *Journal of Computational and Applied Mathematics*, 2014, 259: 503–511. doi: 10.1016/j.cam.2013.06.023.

韦永壮：男，1976年生，教授，博士生导师，研究方向为密码函数、分组密码分析.

史佳利：女，1992年生，硕士生，研究方向为对称密码算法分析.

李灵琛：女，1988年生，博士生，研究方向为分组密码的分析与设计.