分组密码算法uBlock积分攻击的改进

王 晨[①③] 崔佳敏[①③] 李木舟*[①③] 王美琴[①②③]
[①](山东大学网络空间安全学院(研究院) 青岛 266237)
[②](泉城实验室 济南 250100)
[③](山东大学密码技术与信息安全教育部重点实验室 济南 250100)

摘 要：积分攻击是由Daemen等人(doi: 10.1007/BFb0052343)于1997年提出的一种密码分析方法，是继差分分析和线性分析之后最有效的密码分析方法之一。作为2018年全国密码算法设计竞赛分组算法的获胜算法，uBlock抵抗积分攻击的能力受到较多的关注。为了重新评估uBlock家族密码算法抵抗积分攻击的安全性，该文利用单项式传播技术，结合混合整数线性规划(MILP)工具搜索积分区分器，并利用部分和技术进行密钥恢复攻击。对于uBlock-128/128和uBlock-128/256，基于搜索到的9轮积分区分器分别进行了首个11轮和12轮攻击，数据复杂度为2^{127}选择明文，时间复杂度分别为$2^{127.06}$和2^{224}次加密，存储复杂度分别为$2^{44.58}$和 2^{138}字节；对于uBlock-256/256，基于搜索到的10轮积分区分器进行了首个12轮攻击，数据复杂度为2^{253}选择明文，时间复杂度为$2^{253.06}$次加密，存储复杂度为$2^{44.46}$字节。与之前uBlock的最优积分攻击结果相比，uBlock-128/128和uBlock-256/256的攻击轮数分别提高2轮，uBlock-128/256的攻击轮数提高3轮。本文的攻击说明，uBlock针对积分攻击依然有足够的安全冗余。

关键词：密码分析；分组密码；uBlock；积分攻击
中图分类号：TN918.4; TP309.7 **文献标识码**：A **文章编号**：1009-5896(2024)05-0001-10
DOI: 10.11999/JEIT231231

Improved Integral Cryptanalysis on Block Cipher uBlock

WANG Chen[①③] CUI Jiamin[①③] LI Muzhou[①③] WANG Meiqin[①②③]
[①](*School of Cyber Science and Technology*, *Shandong University*, *Qingdao* 266237, *China*)
[②](*Quan Cheng Shandong Laboratory*, *Jinan* 250100, *China*)
[③](*Key Laboratory of Cryptologic Technology and Information Security*, *Ministry of Education*, *Shandong University*, *Jinan* 250100, *China*)

Abstract: Integral attack is one of the most powerful cryptanalytic methods after differential and linear cryptanalysis, which was presented by Daemen et al. in 1997 (doi: 10.1007/BFb0052343). As the winning block cipher of China's National Cipher Designing Competition in 2018, the security strength of uBlock against integral attack has received much attention. To better understand the integral property, this paper constructs the Mixed Integer Linear Programming (MILP) models for monomial prediction to search for the integral distinguishers and uses the partial sum techniques to perform key-recovery attacks. For uBlock-128/128 and uBlock-128/256, this paper gives the first 11 and 12-round attacks based on a 9-round integral distinguisher, respectively. The data complexity is 2^{127} chosen plaintexts. The time complexities are $2^{127.06}$ and 2^{224} times encryptions, respectively. The memory complexities are $2^{44.58}$ and 2^{138} byte, respectively. For uBlock-256/256, this paper gives the first 12-round attack based on a 10-round integral distinguisher. The data complexity is 2^{253} chosen plaintexts. The time and memory complexities are $2^{253.06}$ times encryptions and

收稿日期：2023-11-07；改回日期：2024-01-29；网络出版：2024-03-09
*通信作者： 李木舟 muzhouli@mail.sdu.edu.cn
基金项目：国家重点研发计划(2018YFA0704702)，国家自然科学基金(62032014)，山东省自然科学基金重大基础研究项目(ZR202010220025)，青岛创新项目(QDBSH20230101008)
Foundation Items: The National Key Research and Development Program of China (2018YFA0704702), The National Natural Science Foundation of China (62032014), The Major Basic Research Project of Natural Science Foundation of Shandong Province, China (ZR202010220025), Qingdao Innovation Project (QDBSH20230101008)

$2^{44.46}$ byte, respectively. The number of attacked rounds for uBlock-128/128 and uBlock-256/256 are improved by two rounds compared with the previous best ones. Besides, the number of attacked rounds for uBlock-128/256 is improved by three rounds. The results show that uBlock has enough security margin against integral attack.

Key words: Cryptanalysis; Block cipher; uBlock; Integral attack

1 引言

密码分析是衡量分组密码算法安全性的有效手段，而差分攻击[1,2]和线性攻击[3]是其中两种最早被提出的分析方法。后来陆续有很多新的分析方法被提出，比如积分攻击[4]。1997年，Daemen等人[4]首次提出了积分攻击的原型，用于分析Square算法。在文献[5]中，Knudsen和Wagner正式地给出了积分攻击的概念。积分攻击是一种选择明文的攻击方法。为了构造积分区分器，攻击者首先选择一个明文集合并对其进行r轮加密。如果加密之后得到的一系列状态之和存在某种可预测的积分特性，则称找到了一条r轮积分区分器。最常使用的积分特性为零和特性。当搜索到积分区分器之后，即可以使用该区分器对算法进行密钥恢复。

最初，积分攻击只与算法结构有关。2015年，Todo等人[6]提出一种新的积分性质：可分特性(Division Property)，并对MISTY1算法[7,8]进行了全轮攻击。但可分特性依然是面向字节级运算的，只能利用非线性算法组件的代数次数信息，而无法更精细地利用算法组件的代数结构。为了更加精确地刻画积分性质，Todo等人[9]在2016年提出基于比特的可分特性，包括二子集合比特可分特性和三子集合比特可分特性。但在实际应用中，由于对n-bit算法进行可分特性搜索的时间复杂度约为2^n，因而只能分析分组长度不超过32 bit的算法。2016年，Xiang等人[10]首次提出利用混合整数线性规划(Mixed Integer Linear Programming, MILP)刻画二子集合比特可分特性传播，并借助MILP求解器搜索分组长度大于32 bit的算法积分区分器，多个对称密码算法的积分区分器轮数得以提升[11–14]。2019年，Wang等人[15]发现三子集合比特可分特性可以被用于在cube攻击中恢复超级多项式。在此基础上，Hao等人[16,17]提出了不带未知集合的三子集合比特可分特性。2020年，Hu等人[18]提出单项式预测(Monomial Prediction)技术，从代数角度重新描述了可分特性并证明了单项式预测技术与不带未知集合的三子集合比特可分特性的等价性。借助MILP自动化搜索模型，很多分组密码算法的积分攻击都得到了显著改进[19–21]。

uBlock算法是由吴文玲等人[22]于2018年全国密码算法设计竞赛提出的一个分组密码算法，改进版本于2019年发表在密码学报[23]。uBlock分组长度和主密钥长度都支持128和256-bit，分别为uBlock-128/128，uBlock-128/256和uBlock-256/256。为了表示方便，本文称uBlock-128为分组长度为128的uBlock, uBlock-256为分组长度为256的uBlock。作为2018年全国密码算法设计竞赛分组算法的获胜算法，uBlock抵抗积分攻击的能力受到较多的关注。文献[23]对uBlock不同分组长度算法的积分区分器进行了搜索，结果表明，uBlock-128和uBlock-256分别存在7轮和8轮积分区分器。2020年，Tian等人[24]利用S盒的可分特性传播规则搜索到uBlock更长轮数的积分区分器。其中，uBlock-128存在数据复杂度为2^{124}的8轮积分区分器，而uBlock-256存在数据复杂度为2^{248}的9轮积分区分器。此外，Tian等人[24]首次利用搜索的积分区分器对uBlock-128和uBlock-256分别进行了9轮和10轮密钥恢复。2022年，Mao等人[25]通过分析复杂线性层二子集合比特可分特性传播的非独立性，提出减少冗余可分特性路线的策略，并应用到uBlock-128，从而搜索到更多平衡比特的8轮积分区分器。2023年，黄明等人[26]进一步对复杂线性层进行了研究，并提出一种动态选取可分特性传播技术。通过MILP建模二子集合比特可分特性结合线性层优化技术进行搜索，针对uBlock-128和uBlock-256分别给出了9轮和10轮的积分区分器。

以上的积分区分器搜索均是基于字节级比特可分特性和二子集合比特可分特性，同时在密钥恢复阶段也仅仅外接一轮。因此本文考虑采用最新引入的单项式预测技术重新对uBlock算法的积分区分器进行搜索，同时进一步优化密钥恢复攻击，以全面评估uBlock抵抗积分攻击的能力。具体研究工作如下：

(1)给出了uBlock家族密码的积分区分器搜索结果。本文利用单项式传播技术结合MILP建模针对uBlock加密算法构造搜索模型，所有的轮密钥均被看做独立变量。对于uBlock-128和uBlock-256，分别搜索到最长9轮和10轮的积分区分器；

(2)利用部分和技术给出改进的密钥恢复攻击结果。针对uBlock-128/128，结合等价密钥技术，

本文利用9轮积分区分器给出了首个11轮密钥恢复攻击结果。针对uBlock-128/256，本文根据线性层输入输出依赖关系的相对独立性对密钥恢复攻击流程进行整体优化，从而给出了首个12轮密钥恢复攻击结果。针对uBlock-256/256，利用10轮积分区分器给出了首个12轮密钥恢复攻击结果。具体攻击结果可见表1。

本文结构安排如下：第2节主要介绍了uBlock算法、单项式预测以及部分和技术，第3节主要介绍uBlock算法积分区分器的搜索，第4节详细描述了如何利用部分和技术对uBlock算法进行密钥恢复。第5节对全文的工作进行总结。为了便于复现本文结果，所有的程序都已上传到：https://github.com/Taurus517/uBlock-Integral。

2　基础知识

2.1　分组密码uBlock

uBlock是由吴文玲等人[22,23]提出的一个分组密码算法家族。分组长度和密钥长度都支持128和256-bit。算法共有3个版本，即uBlock-128/128，uBlock-128/256和uBlock-256/256，迭代轮数r分别为16, 24和24。为了方便后文表示，uBlock-128/128和uBlock-128/256统称为uBlock-128，而uBlock-256/256简称uBlock-256。uBlock算法整体采用PX(Pshufb-Xor)结构，PX结构是SPN(Substitution Permutation Network)的细化。如图1所示，uBlock轮函数主要包括3个部分，即轮密钥加，字节替换和线性变换。在r轮迭代后，还有一次额外的轮密钥加操作。假设uBlock算法分组长度为n-bit，主密钥长度为k-bit，接下来具体介绍轮函数的操作：

(1)轮密钥加：记轮函数输入为X_0和X_1。将n-bit轮密钥RK_i平均分为左右两支RK_i^0和RK_i^1，每支为$n/2$-bit，$0 \le i < r$。最后一轮额外的轮密钥记为RK_r。计算$\mathrm{RK}_i^0 \oplus X_0$和$\mathrm{RK}_i^1 \oplus X_1$可以得到字节替换层的输入。

(2)字节替换：字节替换层的左右两支分别由$n/8$个并置的4-bit S盒组成。4-bit S盒的输入输出定义具体可见表2。

假定S盒的输入为(x_0, x_1, x_2, x_3)，输出为(y_0, y_1, y_2, y_3)，则其代数标准型(Algebraic Normal Form, ANF)定义为

$$\begin{cases} y_0 = x_0 \oplus x_1 \oplus x_2 \oplus x_1x_2; \\ y_1 = x_1 \oplus x_2 \oplus x_2x_3 \oplus x_0x_1x_2 \oplus 1; \\ y_2 = x_2 \oplus x_3 \oplus x_0x_1 \oplus x_0x_3 \oplus x_1x_3 \\ \qquad \oplus x_2x_3 \oplus x_1x_2x_3 \oplus 1; \\ y_3 = x_3 \oplus x_0x_1 \oplus 1. \end{cases}$$

由于在密钥恢复过程中，我们使用到的是S盒的逆运算，因此我们同样给出S^{-1}的ANF

$$\begin{cases} x_0 = y_0 \oplus y_1 \oplus y_0y_1 \oplus y_1y_2 \oplus y_1y_3 \oplus y_0y_1y_3 \oplus 1; \\ x_1 = y_1 \oplus y_3 \oplus y_2y_3 \oplus 1; \\ x_2 = y_0 \oplus y_2 \oplus y_3 \oplus y_0y_3; \\ x_3 = y_0 \oplus y_1 \oplus y_0y_1 \oplus y_0y_3 \oplus y_2y_3 \oplus y_0y_2y_3. \end{cases}$$

(3)线性变换：线性变换层包含异或，每32 bits循环左移j bit，以及字节置换3种操作。具体过程为

$$\begin{aligned} X_1 &\leftarrow X_1 \oplus X_0 \\ X_0 &\leftarrow X_0 \oplus \left(X_1 \underset{32}{\lll} 4\right) \\ X_1 &\leftarrow X_1 \oplus \left(X_0 \underset{32}{\lll} 8\right) \\ X_0 &\leftarrow X_0 \oplus \left(X_1 \underset{32}{\lll} 8\right) \\ X_1 &\leftarrow X_1 \oplus \left(X_0 \underset{32}{\lll} 32\right) \\ X_0 &\leftarrow X_0 \oplus X_1 \\ X_0 &\leftarrow \mathrm{PL_n}\left(X_0\right) \\ X_1 &\leftarrow \mathrm{PR_n}\left(X_1\right) \end{aligned}$$

表 1　uBlock积分分析结果比较

算法版本	攻击轮数	积分区分器	数据复杂度	时间复杂度	存储复杂度	参考文献
uBlock-128/128	9	$(\mathcal{C}^4, \mathcal{A}^{124}) \overset{8\mathrm{R}}{\to} (\mathcal{B}^1, \mathcal{U}^2, \mathcal{B}^1)^{32}$	2^{124}	$2^{125.47}$	$2^{6.25}$	[24]
	10	$(\mathcal{C}^4, \mathcal{A}^{124}) \overset{8\mathrm{R}}{\to} (\mathcal{B}^1, \mathcal{U}^1, \mathcal{B}^2)^{32}$	2^{124}	$2^{124.07}$	$2^{44.32}$	本文
	11	$(\mathcal{A}^1, \mathcal{C}^1, \mathcal{A}^{126}) \overset{9\mathrm{R}}{\to} (\mathcal{U}^3, \mathcal{B}^1)^{32}$	2^{127}	$2^{127.06}$	$2^{44.58}$	本文
uBlock-128/256	9	$(\mathcal{C}^4, \mathcal{A}^{124}) \overset{8\mathrm{R}}{\to} (\mathcal{B}^1, \mathcal{U}^2, \mathcal{B}^1)^{32}$	2^{124}	$2^{125.47}$	$2^{6.25}$	[24]
	11	$(\mathcal{C}^4, \mathcal{A}^{124}) \overset{8\mathrm{R}}{\to} (\mathcal{B}^1, \mathcal{U}^1, \mathcal{B}^2)^{32}$	2^{124}	$2^{179.19}$	$2^{137.81}$	本文
	11	$(\mathcal{A}^1, \mathcal{C}^1, \mathcal{A}^{126}) \overset{9\mathrm{R}}{\to} (\mathcal{U}^3, \mathcal{B}^1)^{32}$	2^{127}	2^{224}	2^{46}	本文
	12	$(\mathcal{A}^1, \mathcal{C}^1, \mathcal{A}^{126}) \overset{9\mathrm{R}}{\to} (\mathcal{U}^3, \mathcal{B}^1)^{32}$	2^{127}	2^{224}	2^{138}	本文
uBlock-256/256	10	$(\mathrm{C}^8, \mathcal{A}^{248}) \overset{9\mathrm{R}}{\to} (\mathcal{B}^1, \mathcal{U}^2, \mathcal{B}^1)^{64}$	2^{248}	$2^{249.38}$	$2^{7.25}$	[24]
	11	$(\mathcal{C}^8, \mathcal{A}^{248}) \overset{9\mathrm{R}}{\to} (\mathcal{B}^1, \mathcal{U}^2, \mathcal{B}^1)^{64}$	2^{248}	$2^{248.06}$	$2^{44.32}$	本文
	12	$(\mathcal{C}^3, \mathcal{A}^{253}) \overset{10\mathrm{R}}{\to} (\mathcal{U}^3, \mathcal{B}^1)^{64}$	2^{253}	$2^{253.06}$	$2^{44.46}$	本文

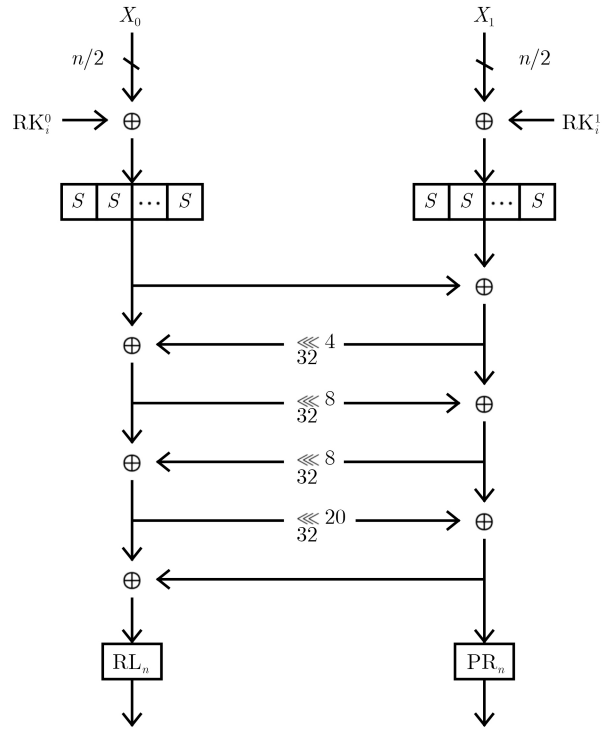

图 1 uBlock轮函数

字节置换定义如表3所示。其中，PL_{128}和PR_{128}应用于uBlock-128，PL_{256}和PR_{256}应用于uBlock-256。这里以PL_{128}举例

$$(y_0,y_1,y_2,y_3,y_4,y_5,y_6,y_7)\overset{\mathrm{PL}_{128}}{\rightarrow}(y_1,y_3,y_4,y_6,y_0,y_2,y_7,y_5),$$

其中$y_i\in\{0,1\}^8$, $0\le i<8$。

2.2 单项式预测与MILP建模

2.2.1 单项式预测

单向式预测技术是由Hu等人[18]在亚密会2020(ASIACRYPT 2020)提出，从多项式的角度重新解释了可分特性。首先定义向量布尔函数$\boldsymbol{f}:\mathbb{F}_2^{n_0}\to\mathbb{F}_2^{n_r}$, $\boldsymbol{y}=\boldsymbol{f}(\boldsymbol{x})$。$\boldsymbol{f}$是由多个向量布尔函数$\boldsymbol{f}^{(i)}:\mathbb{F}_2^{n_i}\to\mathbb{F}_2^{n_{i+1}}$, $\boldsymbol{x}^{(i+1)}=\boldsymbol{f}^{(i)}\left(\boldsymbol{x}^{(i)}\right)$, $0\le i<r$，组成的复合向量布尔函数，即

$$\boldsymbol{y}=\boldsymbol{f}(\boldsymbol{x})=\boldsymbol{f}^{(r-1)}\circ\boldsymbol{f}^{(r-2)}\circ\cdots\circ\boldsymbol{f}^{(0)}$$

本文使用$\pi_{\boldsymbol{u}}(\boldsymbol{x})$表示$\mathbb{F}_2^n$上的一个单项式，即$\pi_{\boldsymbol{u}}(\boldsymbol{x})=\prod_{i=0}^{n-1}x_i^{u_i}$。其中，$\boldsymbol{x}=(x_0,\cdots,x_{n-1})\in\mathbb{F}_2^n$, $\boldsymbol{u}=(u_0,\cdots,u_{n-1})\in\mathbb{F}_2^n$。在密码学中，由于$\boldsymbol{f}$的ANF通常非常复杂，以至于无法计算或存储。因此判断某个给定的单项式$\pi_{\boldsymbol{u}^{(0)}}\left(\boldsymbol{x}^{(0)}\right)$是否存在于$\pi_{\boldsymbol{u}^{(r)}}\left(\boldsymbol{x}^{(r)}\right)$通常是一个困难问题。但通过复合函数的性质，可以将这个大问题进行分解。如果已知$\boldsymbol{f}^{(0)}$的ANF，那么从$\boldsymbol{x}^{(0)}$的一个单项式$\pi_{\boldsymbol{u}^{(0)}}\left(\boldsymbol{x}^{(0)}\right)$出发，可以很容易的得到所有包含$\pi_{\boldsymbol{u}^{(0)}}\left(\boldsymbol{x}^{(0)}\right)$的$\pi_{\boldsymbol{u}^{(1)}}\left(\boldsymbol{x}^{(1)}\right)$；同理，若已知$\boldsymbol{f}^{(1)}$的ANF，可以得到所有包含$\pi_{\boldsymbol{u}^{(1)}}\left(\boldsymbol{x}^{(1)}\right)$的$\pi_{\boldsymbol{u}^{(2)}}\left(\boldsymbol{x}^{(2)}\right)$；以此类推直至$\pi_{\boldsymbol{u}^{(r)}}\left(\boldsymbol{x}^{(r)}\right)$。令$\pi_{\boldsymbol{u}^{(i)}}\left(\boldsymbol{x}^{(i)}\right)\to\pi_{\boldsymbol{u}^{(i+1)}}\left(\boldsymbol{x}^{(i+1)}\right)$表示$\pi_{\boldsymbol{u}^{(i+1)}}\left(\boldsymbol{x}^{(i+1)}\right)$包含$\pi_{\boldsymbol{u}^{(i)}}\left(\boldsymbol{x}^{(i)}\right)$，下面给出单项式路径(Monomial Trail)的定义。

定义1(单项式路径[18])。定义复合函数$\boldsymbol{f}=\boldsymbol{f}^{(r-1)}\circ\boldsymbol{f}^{(r-2)}\circ\cdots\circ\boldsymbol{f}^{(0)}$，其组成函数的输入输出为$\boldsymbol{x}^{(i+1)}=\boldsymbol{f}^{(i)}\left(\boldsymbol{x}^{(i)}\right)$，$0\le i<r$。如果一个单项式序列$\left(\pi_{\boldsymbol{u}^{(0)}}(\boldsymbol{x}^{(0)}),\pi_{\boldsymbol{u}^{(1)}}(\boldsymbol{x}^{(1)}),\cdots,\pi_{\boldsymbol{u}^{(r)}}(\boldsymbol{x}^{(r)})\right)$满足$\pi_{\boldsymbol{u}^{(0)}}\left(\boldsymbol{x}^{(0)}\right)\to\cdots\to\pi_{\boldsymbol{u}^{(i)}}\left(\boldsymbol{x}^{(i)}\right)\to\cdots\to\pi_{\boldsymbol{u}^{(r)}}\left(\boldsymbol{x}^{(r)}\right)$，则被称为布尔函数$\boldsymbol{f}$连接$\pi_{\boldsymbol{u}^{(0)}}\left(\boldsymbol{x}^{(0)}\right)$和$\pi_{\boldsymbol{u}^{(r)}}\left(\boldsymbol{x}^{(r)}\right)$的$r$轮单项式路径。如果至少有一条单项式路径连接$\pi_{\boldsymbol{u}^{(0)}}\left(\boldsymbol{x}^{(0)}\right)$和$\pi_{\boldsymbol{u}^{(r)}}\left(\boldsymbol{x}^{(r)}\right)$，则记作$\pi_{\boldsymbol{u}^{(0)}}\left(\boldsymbol{x}^{(0)}\right)\nrightarrow\pi_{\boldsymbol{u}^{(r)}}\left(\boldsymbol{x}^{(r)}\right)$。否则，记作$\pi_{\boldsymbol{u}^{(0)}}\left(\boldsymbol{x}^{(0)}\right)\nrightarrow\pi_{\boldsymbol{u}^{(r)}}\left(\boldsymbol{x}^{(r)}\right)$。

在文献[18]中，通过判断单项式路径数目的奇偶性，可以严格地判断一个单项式是否存在于最终多项式中。但对于分组密码来说，单项式路径数量往往很大，数量的计算是一件困难的事。因此，一些在计算效率和精确度上的折中有时是必须的。在本文中，主要采取以下引理来进行积分区分器的搜索。

引理1(文献[18])。如果$\pi_{\boldsymbol{u}^{(0)}}\left(\boldsymbol{x}^{(0)}\right)\to\pi_{\boldsymbol{u}^{(r)}}\left(\boldsymbol{x}^{(r)}\right)$，则$\pi_{\boldsymbol{u}^{(0)}}\left(\boldsymbol{x}^{(0)}\right)\nrightarrow\pi_{\boldsymbol{u}^{(r)}}\left(\boldsymbol{x}^{(r)}\right)$，而$\pi_{\boldsymbol{u}^{(0)}}\left(\boldsymbol{x}^{(0)}\right)\nrightarrow\pi_{\boldsymbol{u}^{(r)}}\left(\boldsymbol{x}^{(r)}\right)$，则表示$\pi_{\boldsymbol{u}^{(0)}}\left(\boldsymbol{x}^{(0)}\right)\nrightarrow\pi_{\boldsymbol{u}^{(r)}}\left(\boldsymbol{x}^{(r)}\right)$。

表 2 4-bit S盒(S)

x	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
$S(x)$	0x7	0x4	0x9	0xc	0xb	0xa	0xd	0x8	0xf	0xe	0x1	0x6	0x0	0x3	0x2	0x5

表 3 PL_n和PR_n

类型	置换后
PL_{128}	{1, 3, 4, 6, 0, 2, 7, 5}
PR_{128}	{2, 7, 5, 0, 1, 6, 4, 3}
PL_{256}	{2, 7, 8, 13, 3, 6, 9, 12, 1, 4, 15, 10, 14, 11, 5, 0}
PR_{256}	{6, 11, 1, 12, 9, 4, 2, 15, 7, 0, 13, 10, 14, 3, 8, 5}

2.2.2 单项式预测的MILP建模

接下来介绍如何用MILP刻画单项式路径的传播。考虑单项式路径$(\pi_{\boldsymbol{u}^{(0)}}(\boldsymbol{x}^{(0)}),\pi_{\boldsymbol{u}^{(1)}}(\boldsymbol{x}^{(1)}),\cdots,\pi_{\boldsymbol{u}^{(r)}}(\boldsymbol{x}^{(r)}))$，由于$\boldsymbol{x}^{(i)}$是符号变量，因此只需要刻画$\boldsymbol{u}^{(0)},\boldsymbol{u}^{(1)},\cdots,\boldsymbol{u}^{(r)}$的传播。对于任意布尔函数$\boldsymbol{y}=\boldsymbol{f}(x)$，只要$\boldsymbol{x}^{\boldsymbol{u}}\to\boldsymbol{y}^{\boldsymbol{v}}$，则$(\boldsymbol{u},\boldsymbol{v})$是$\boldsymbol{f}$的有效单项式路径。由于分组密码算法可以分解为许多小组件，如异或，复制，S盒以及线性层等，为了搜索积分区分器，只需要按照各个组件的传播规则对$(\boldsymbol{u},\boldsymbol{v})$进行约束。下面介绍搜索uBlock积分区分器所需要的4种传播模型。

模型1(复制(COPY)[16,17])。定义$(a)\overset{\text{COPY}}{\to}(b_1,b_2,\cdots,b_n)$表示复制函数的单项式路径，即1 bit变量复制成$n$bit变量。具体可用如下MILP语句进行约束

$$\begin{cases}b_1+b_2+\cdots+b_n\geq a;\\ a\geq b_i,\text{for all}i\in\{1,2,\cdots,n\};\\ a,b_1,b_2,\cdots,b_n\text{是二进制变量.}\end{cases}$$

模型2(异或(XOR)[16,17])。定义$(a_1,a_2,\cdots,a_n)\overset{\text{XOR}}{\to}(b)$表示异或函数的单项式路径，即$n$bit变量经过异或运算后得1 bit变量。具体可用如下MILP语句进行约束

$$\begin{cases}a_1+a_2+\cdots+a_n-b=0;\\ a_1,a_2,\cdots,a_n,b\text{是二进制变量.}\end{cases}$$

模型3(S盒[10,25])。考虑一个$n$bit进$m$bit出的S盒，则S盒的单项式路径可表示成$(n+m)$维的二进制向量。借助Sagemath[27]的inequality_generator()函数，可以得到一系列不等式。最后借助贪心算法[28]对不等式数量进行约减。

模型4(线性层[16,29])。线性层$\boldsymbol{M}$是一个$n\times n$的矩阵，即

$$\boldsymbol{M}=\begin{pmatrix}m_{0,0}&m_{0,1}&\cdots&m_{0,n-1}\\ m_{1,0}&m_{1,1}&\cdots&m_{1,n-1}\\ \vdots&\vdots&\ddots&\vdots\\ m_{n-1,0}&m_{n-1,1}&\cdots&m_{n-1,n-1}\end{pmatrix},$$

其中$m_{i,j}\in\{0,1\}$。为了表示$\boldsymbol{M}$的单项式路径，需要引入二进制变量$t_{i,j}$，$0\leq i,j<n$。定义$(x_0,x_1,\cdots,x_{n-1})\to(y_0,y_1,\cdots,y_{n-1})$表示$\boldsymbol{M}$的单项式路径，借助模型1和模型2，可以刻画线性层的MILP模型，即

$x_j\overset{\text{COPY}}{\to}(t_{0,j},t_{1,j},\cdots,t_{n-1,j})$和$(t_{i,0},t_{i,1},\cdots,t_{i,n-1})\overset{\text{XOR}}{\to}y_i$。

2.3 部分和

在密钥恢复过程中采用部分和[30]技术能够降低攻击复杂度。下面举例介绍部分和技术。假设有密文集合$\mathbb{C}=\{c_i|0\leq i<m\}$。定义$c_{i,j}$为$c_i$的第$j$个半字节，$c_{i,j}\in\mathbb{F}_2^4$，$0\leq j<n$。假设希望计算$x=\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}S^{-1}(c_{i,j}\oplus k_j)$，其中，$k=(k_0,\cdots,k_{n-1})$表示需要猜测的$\boldsymbol{n}$个半字节密钥。为了计算$x$，首先将密文存放在长为$4n$ bit的计数器中，然后猜测k_0,k_1的值并计算$x=\sum_{i=0}^{m-1}(S^{-1}(c_{i,0}\oplus k_0)\oplus S^{-1}(c_{i,1}\oplus k_1))$。此时计数器变为$(4n-4)$bit。接着依次猜测其余$k_j$的值并计算$x=x\oplus\sum_{i=0}^{m-1}S^{-1}(c_{i,j}\oplus k_j)$，$2\leq j<n$。为了计算$x$，总共需要执行1次8-bit密钥猜测以及$(n-2)$次4-bit密钥猜测。因此总共需要查询$2^{4n}\times2^8\times2+2^{4n-4}\times2^8\times2^4+2^{4n-8}\times2^8\times2^4\times2^4+\ldots+2^{4n-4\times(n-2)}\times2^8\times2^{4\times(n-2)}=2^{4n}\times2^8\times n$次S盒。

3 uBlock自动化搜索模型以及积分区分器

3.1 uBlock的MILP建模

假设uBlock的分组长度为n-bit。当$0\leq i<r$时，$\pi_{\boldsymbol{u}^{(i)}}(\boldsymbol{x}^{(i)})$和$\pi_{\boldsymbol{u}^{(i+1)}}(\boldsymbol{x}^{(i+1)})$分别表示第$i$轮轮函数输入和输出状态的单项式。其中，$\pi_{\boldsymbol{u}^{(0)}}(\boldsymbol{x}^{(0)})$表示明文处的单项式，$\pi_{\boldsymbol{u}^{(r)}}(\boldsymbol{x}^{(r)})$则表示密文处的单项式。$\pi_{\boldsymbol{v}^{(i)}}(\boldsymbol{k}^{(i)})$表示第$i$轮轮密钥的单项式。首先将uBlock的轮函数拆解成基本操作，利用2.2.2节介绍的模型对$\boldsymbol{u}^{(i)}$和$\boldsymbol{v}^{(i)}$添加约束条件。值得注意的是，所有的轮密钥均看做是独立变量。接着，对模型添加初始和终止条件。

(1)初始条件：给定一个明文结构。假设明文的所有活跃比特为集合$I\subset\{0,1,\cdots,n-1\}$，按如下规则对$\boldsymbol{u}^{(0)}$添加初始条件

$$\begin{cases}u_i^{(0)}=1,\text{若}i\in I\\ u_i^{(0)}=0,\text{若}i\notin I\end{cases}$$

对于$(\boldsymbol{v}^{(0)},\boldsymbol{v}^{(1)},\cdots,\boldsymbol{v}^{(r)})$，不需要添加任何约束。

(2)终止条件：假如要考虑密文第i'个比特的积分性质，则需要按如下规则对$\boldsymbol{u}^{(r)}$添加终止条件

$$\begin{cases}u_i^{(r)}=1,\text{若}i=i'\\ u_i^{(r)}=0,\text{若}i\neq i'\end{cases}$$

当搜索uBlock积分区分器的MILP模型刻画完成，便可以调用MILP求解器对模型进行求解。如果模型无解，对任意$\pi_{\boldsymbol{v}^{(0)},\cdots,\boldsymbol{v}^{(r)}}(\boldsymbol{k}^{(0)},\cdots,\boldsymbol{k}^{(r)})\cdot\pi_{\boldsymbol{u}^{(0)}}(\boldsymbol{x}^{(0)})$，均有$\pi_{\boldsymbol{v}^{(0)},\cdots,\boldsymbol{v}^{(r)}}(\boldsymbol{k}^{(0)},\cdots,\boldsymbol{k}^{(r)})\cdot\pi_{\boldsymbol{u}^{(0)}}(\boldsymbol{x}^{(0)})\nrightarrow\pi_{\boldsymbol{u}^{(r)}}(\boldsymbol{x}^{(r)})$，则密文的第$i'$个比特具有零和性质；如果模型有解，说明$\pi_{\boldsymbol{v}^{(0)},\cdots,\boldsymbol{v}^{(r)}}(\boldsymbol{k}^{(0)},\cdots,\boldsymbol{k}^{(r)})\cdot\pi_{\boldsymbol{u}^{(0)}}(\boldsymbol{x}^{(0)})\nrightarrow\pi_{\boldsymbol{u}^{(r)}}(\boldsymbol{x}^{(r)})$，密文第$i'$个比特的平衡性无法确定。

3.2 积分区分器

本文目标是搜索uBlock最长轮积分区分器，对于uBlock-128，设置初始条件中$\boldsymbol{u}^{(0)}$的1 bit为常数，其余127 bit活跃，通过遍历初始条件中常数比特的位置来搜索r轮积分区分器。如果密文处出现平衡比特，则轮数增加到$r+1$，继续重复上述搜索过程直到没有平衡比特出现。接着，在最长轮积分区分器的基础上，通过增加初始条件中常数比特的个数来减少数据复杂度。因为uBlock算法为半字节运算，所以可以优先考虑同一半字节内的常数比特位置。按照上述方法，本文搜索到uBlock-128最长9轮积分区分器，数据复杂度为2^{127}选择明文。uBlock-256积分区分器的搜索过程与uBlock-128类似，搜索到最长10轮积分区分器，数据复杂度为2^{253}选择明文。具体的uBlock算法积分区分器如下所示。其中$\mathcal{C}$表示常数比特，$\mathcal{A}$表示活跃比特，$\mathcal{B}$表示平衡比特，$\mathcal{U}$表示未知比特

(1)uBlock-128的8轮和9轮积分区分器分别为

$$(\mathcal{C}^4,\mathcal{A}^{60},\mathcal{A}^{64}) \overset{8\mathrm{R}}{\to} ((\mathcal{B}^1,\mathcal{U}^2,\mathcal{B}^1)^{16},(\mathcal{B}^1,\mathcal{U}^2,\mathcal{B}^1)^{16})$$

$$(\mathcal{A}^1,\mathcal{C}^1,\mathcal{A}^{62},\mathcal{A}^{64}) \overset{9\mathrm{R}}{\to} ((\mathcal{U}^3,\mathcal{B}^1)^{16},(\mathcal{U}^3,\mathcal{B}^1)^{16})$$

(2)uBlock-256的9轮和10轮积分区分器分别为

$$(\mathcal{C}^8,\mathcal{A}^{120},\mathcal{A}^{128}) \overset{9\mathrm{R}}{\to} ((\mathcal{B}^1,\mathcal{U}^2,\mathcal{B}^1)^{32},(\mathcal{B}^1,\mathcal{U}^2,\mathcal{B}^1)^{32})$$

$$(\mathcal{C}^3,\mathcal{A}^{125},\mathcal{A}^{128}) \overset{10\mathrm{R}}{\to} ((\mathcal{U}^3,\mathcal{B}^1)^{32},(\mathcal{U}^3,\mathcal{B}^1)^{32})$$

4 uBlock密钥恢复

基于3.2节搜索到的积分区分器，本节利用部分和技术对uBlock算法进行密钥恢复。对于uBlock-128/128和uBlock-128/256，基于相同的9轮积分区分器分别进行了11轮和12轮的密钥恢复。而对于uBlock-256/256，则基于10轮积分区分器进行了12轮攻击。受篇幅限制，本文在4.1节和4.2节分别具体介绍uBlock-128/128的11轮和uBlock-128/256的12轮积分攻击。uBlock-256/256的攻击过程与uBlock-128/128类似。

4.1 11轮uBlock-128/128密钥恢复攻击

本节基于9轮积分区分器$(\mathcal{A}^1,\mathcal{C}^1,\mathcal{A}^{62},\mathcal{A}^{64})\to((\mathcal{U}^3,\mathcal{B}^1)^{16},(\mathcal{U}^3,\mathcal{B}^1)^{16})$，并在区分器尾部添加两轮，最终得到uBlock-128/128算法11轮密钥恢复攻击。初始加密流程如

$$\mathrm{Dis.}\to x_9 \overset{S_9}{\to} y_9 \overset{L_9}{\to} z_9 \overset{\mathrm{RK}_{10}}{\to} x_{10} \overset{S_{10}}{\to} y_{10} \overset{L_{10}}{\to} z_{10} \overset{\mathrm{RK}_{11}}{\to} C$$

其中，Dis.代表9轮积分区分器，S_i,L_i和RK_{i+1}分别代表第i轮轮函数中的字节替换，线性变换以及轮密钥加，C代表密文，x_i,y_i,z_i分别代表S_i，L_i和RK_{i+1}运算的输入状态。利用等价密钥技术，可以将RK_{i+1}和L_i的位置进行调换，则加密流程变为

$$\mathrm{Dis.}\to x_9 \overset{S_9}{\to} y_9 \overset{\mathrm{EK}_{10}}{\to} z_9 \overset{L_9}{\to} x_{10} \overset{S_{10}}{\to} y_{10} \overset{\mathrm{EK}_{11}}{\to} z_{10} \overset{L_{10}}{\to} C$$

其中，$\mathrm{EK}_i=L^{-1}(\mathrm{RK}_i)$为等价密钥。

为了计算平衡比特$x_9[3]$处的值，本文将整体计算分解为两个步骤，分别推算出所需的密钥比特：

(1)推算由z_9计算$x_9[3]$需要猜测的密钥比特。

由S盒的ANF可知，

$$\begin{aligned}x_9[3] =& y_9[0]\oplus y_9[1]\oplus y_9[0]\cdot y_9[1]\oplus y_9[0]\cdot y_9[3]\\ &\oplus y_9[2]\cdot y_9[3]\oplus y_9[0]\cdot y_9[2]\cdot y_9[3]\\ =&(z_9[0]\oplus\mathrm{EK}_{10}[0])\oplus(z_9[1]\oplus\mathrm{EK}_{10}[1])\\ &\oplus(z_9[0]\oplus\mathrm{EK}_{10}[0])\cdot(z_9[1]\oplus\mathrm{EK}_{10}[1])\\ &\oplus(z_9[0]\oplus\mathrm{EK}_{10}[0])\cdot(z_9[3]\oplus\mathrm{EK}_{10}[3])\\ &\oplus(z_9[2]\oplus\mathrm{EK}_{10}[2])\cdot(z_9[3]\oplus\mathrm{EK}_{10}[3])\\ &\oplus(z_9[0]\oplus\mathrm{EK}_{10}[0])\cdot(z_9[2]\oplus\mathrm{EK}_{10}[2])\\ &\cdot(z_9[3]\oplus\mathrm{EK}_{10}[3])\end{aligned}$$

如果已知$z_9[0\sim3]$的值，通过猜测$\mathrm{EK}_{10}[0\sim3]$，便可计算出$x_9[3]$处的值。

(2)推算由z_{10}计算$z_9[0\sim3]$需要猜测的密钥比特。

由轮函数可知，

$$\begin{aligned}z_9[0\sim3] =& (x_{10}[0],x_{10}[1],\cdots,x_{10}[127])^{\mathrm{T}}\cdot L^{-1}[0\sim3]\\ =&\sum_{i\in I}(x_{10}[i],x_{10}[i+1],x_{10}[i+2],\\ &x_{10}[i+3])\\ =&\sum_{i\in I}(S^{-1}(y_{10}[i]),S^{-1}(y_{10}[i+1]),\\ &S^{-1}(y_{10}[i+2]),S^{-1}(y_{10}[i+3]))\\ =&\sum_{i\in I}(S^{-1}(z_{10}[i]\oplus\mathrm{EK}_{11}[i]),\\ &S^{-1}(z_{10}[i+1]\oplus\mathrm{EK}_{11}[i+1]),\\ &S^{-1}(z_{10}[i+2]\oplus\mathrm{EK}_{11}[i+2]),\\ &S^{-1}(z_{10}[i+3]\oplus\mathrm{EK}_{11}[i+3]))\end{aligned}$$

如果已知$z_{10}[i\sim(i+3)]$的值，通过猜测$\mathrm{EK}_{11}[i\sim(i+3)]$，便可计算出$z_9[0\sim3]$的值，其中，$i\in\{0,4,8,12,32,68,88,92,100,120,124\}$。

综上所述，为了计算$x_9[3]$，共需要猜测48 bit密钥，具体过程如图2所示。

利用部分和技术，下面我们给出密钥恢复的具体过程：

(1)准备计数器。

(a)选择2^{127}形式为$(\mathcal{A}^1,C^1,\mathcal{A}^{62},\mathcal{A}^{64})$的明文集合$\mathbb{P}$。

(b)分配大小为2^{44}的计数器T_0，初始化为0。

对明文集合$\mathbb{P}$中的每一个明文，分别进行11轮加密得到对应的密文C，然后执行L_{10}^{-1}操作，得

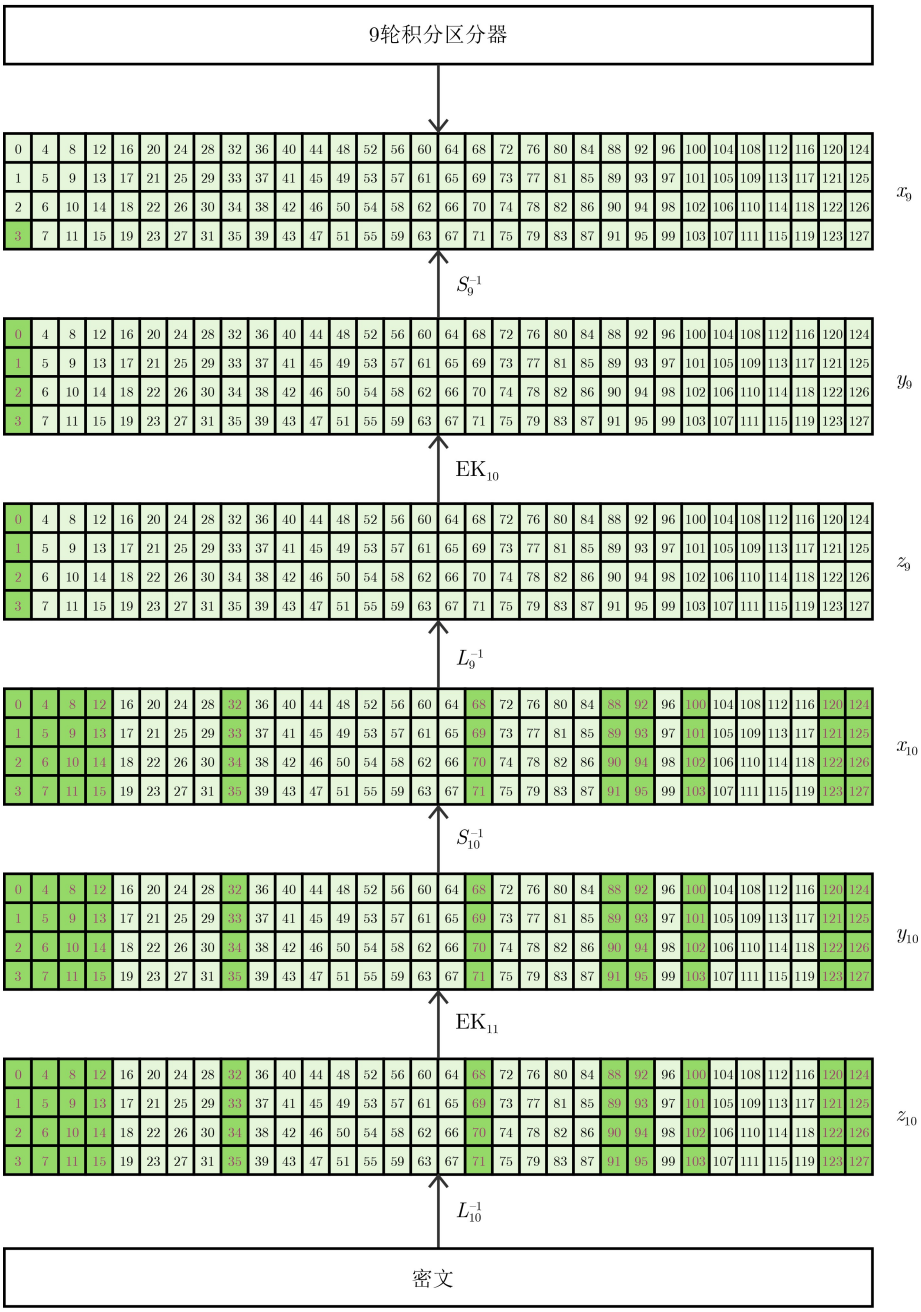

图 2 uBlock-128/128算法11轮密钥恢复

z_{10}。T_0以44-bit z_{10}的级联值为索引i，$i = z_{10}[0\sim15] \| z_{10}[32\sim35] \| z_{10}[68\sim71] \| z_{10}[88\sim95] \| z_{10}[100\sim103] \| z_{10}[120\sim127]$。根据44-bit索引值对$T_0$进行更新，即$T_0[i] = T_0[i] + 1 \bmod 2$。

(2)猜测44-bit $\mathrm{EK}_{11}[i\sim(i+3)]$，$i \in \{0,4,8,12,32,68,88,92,100,120,124\}$。

针对每个4-bit猜测，部分解密z_{10}到x_{10}，并计算$z_9[0\sim3]$对应4-bit x_{10}的分量。此时计数器由44 bit变为4 bit。该部分所需要的时间复杂度为$2^{44} \times 2^8 \times 11 \approx 2^{55.46}$次4-bit S盒查询和$2^{55.46}/11 \approx 2^{52}$次4-bit线性计算。

(3)猜测4-bit $\mathrm{EK}_{10}[0\sim3]$。

对于每一个猜测，计算$x_9[3]$的对应值。保留让$x_9[3] = 0$的密钥为正确密钥。该部分所需要的时间复杂度为$2^4 \times 2^{44} \times 2^4 = 2^{52}$次4-bit S盒查询。

(4)穷搜剩余密钥。

每次猜测48-bit密钥后，128-bit的密钥空间都缩小1/2。因此只需要选取区分器中不同的平衡比特位置，重复12次上述操作即可，最后遍历剩余的2^{128-12}密钥空间。

攻击复杂度：本文将字节替换和线性操作分别看作半轮加密，则

步骤(1)的时间复杂度为$2^{127}+2^{127}/2/11$次11轮加密；

步骤(2)和(3)的时间复杂度为T_1+T_2，其中$T_1=(2^{55.46}+2^{52})/32$次128-bit S盒查询，$T_2=2^{52}/32$次128-bit线性计算。

步骤(4)的时间复杂度为$12\times(T_1+T_2)/2/11+2^{116}\approx 2^{116}$次11轮加密。

综上所述，攻击11轮uBlock-128/128所需要的总时间复杂度为$T=2^{127}+2^{127}/2/11+2^{116}\approx 2^{127.06}$次11轮加密。内存复杂度为$M=(2^{44}\times 12)/8\approx 2^{44.58}$字节。数据复杂度为$2^{127}$选择明文。

4.2 12轮uBlock-128/256密钥恢复攻击

由于uBlock-128/128和uBlock-128/256之间的区别仅为密钥生成算法的不同，因此uBlock-128/256的11轮密钥恢复攻击与uBlock-128/128类似。在11轮密钥恢复攻击的基础上，本文在尾部再额外添加一轮，用于进行uBlock-128/256算法的密钥恢复攻击。同样的，利用等价密钥技术，可以将RK_{i+1}和L_i的位置进行调换，加密流程为

$$\begin{aligned}&\text{Dis.}\rightarrow x_9\overset{S_9}{\rightarrow}y_9\overset{\mathrm{EK}_{10}}{\rightarrow}z_9\overset{L_9}{\rightarrow}x_{10}\overset{S_{10}}{\rightarrow}y_{10}\\&\overset{\mathrm{EK}_{11}}{\rightarrow}z_{10}\overset{L_{10}}{\rightarrow}x_{11}\overset{S_{11}}{\rightarrow}y_{11}\overset{\mathrm{EK}_{12}}{\rightarrow}z_{11}\overset{L_{11}}{\rightarrow}C。\end{aligned}$$

但是在12轮密钥恢复攻击中，最后一轮达到了全扩散，即计算1-bit积分区分器尾部值需要用到128-bitz_{11}。

通过分析线性层逆矩阵的关系，发现线性层可以划分为两个独立的部分分别计算。其中，输入的第32～63和96～127 bit仅与输出的第16～31, 48～63, 72～87和104～119 bit有关；而输入的第0～31和64～95 bit仅与输出的第0～15, 32～47, 64～71, 88～103和120～127 bit有关。因此，z_{10}的第i_0～(i_0+3) bit仅与x_{11}的第i_1～(i_1+3) bit有关，而z_{10}的第i_2～(i_2+3) bit仅与x_{11}的第 i_3～(i_3+3) bit有关。其中$i_0\in\{32,100,120,124\}$, $i_1\in$\{16, 20, 24, 28, 48, 52, 56, 60, 72, 76, 80, 84, 104, 108, 112, 116\}，$i_2\in$ \{0, 4, 8, 12, 68, 88, 92\}，$i_3\in$ \{0, 4, 8, 12, 32, 36, 40, 44, 64, 68, 88, 92, 96, 100, 120, 124\}。结合线性层的相对独立性，本文可以对密钥恢复攻击流程进行优化。改进后的密钥恢复过程如下：

(1)准备计数器。

(a)选择2^{127}形式为$(\mathcal{A}^1,C^1,\mathcal{A}^{62},\mathcal{A}^{64})$的明文集合$\mathbb{P}$。

(b)分配大小为2^{128}的计数器T_0，初始化为0。

对明文集合$\mathbb{P}$中的每一个明文，分别进行12轮加密得到对应的密文C，然后执行L_{11}^{-1}操作，得z_{11}。T_0以128-bit$z_{11}[0\sim127]$的值为索引i。根据128-bit索引值对T_0进行更新，即$T_0[i]=T_0[i]+1\bmod 2$。

(2)猜128-bitEK_{12}。

(a)猜测8-bit$\mathrm{EK}_{12}[20\sim23],\mathrm{EK}_{12}[72\sim75]$。

针对每个4-bit猜测，部分解密z_{11}到x_{11}，并计算$z_{10}[i_0\sim(i_0+3)]$对应4-bitx_{11}的分量。此时，计数器由128 bits变为136 bits。该部分的时间复杂度为$(2^{128}\times 2^4+2^{132}\times 2^4\times 2^4)\approx 2^{140}$次4-bit S盒查询以及$2^{140}\times 2/11\approx 2^{137.55}$次4-bit线性计算。

(b)基于已猜测的8 bits密钥，猜测56-bit $\mathrm{EK}_{12}[i'\sim(i'+3)]$，$i'\in i_1/\{20,72\}$。

针对每个4-bit猜测，部分解密z_{11}到x_{11}，并计算$z_{10}[i_0\sim(i_0+3)]$对应4-bitx_{11}的分量。此时，计数器由136 bits变为80 bits。该部分的时间复杂度为$2^8\times(2^{136}\times 2^4\times 14)\approx 2^{151.81}$次4-bit S盒查询以及$2^{151.81}\times 4/11\approx 2^{150.35}$次4-bit线性计算。

(c)基于已猜测的64 bits密钥，猜测8-bit $\mathrm{EK}_{12}[4\sim7],\mathrm{EK}_{11}[32\sim35]$。

针对每个4-bit猜测，部分解密z_{11}到x_{11}，并计算$z_{10}[i_2\sim(i_2+3)]$对应4-bitx_{11}的分量。此时，计数器由80 bits变为100 bits。该部分的时间复杂度为$2^{64}\times(2^{80}\times 2^4+2^{92}\times 2^4\times 2^4)\approx 2^{164}$次4-bit S盒查询以及$2^{164}\times 4/11\approx 2^{162.54}$次4-bit线性计算。

(d)基于已猜测的72 bits密钥，猜测56-bit $\mathrm{EK}_{12}[i'\sim(i'+3)]$，$i'\in i_3/\{4,32\}$。

针对每个4-bit猜测，部分解密z_{11}到x_{11}，并计算$z_{10}[i_2\sim(i_2+3)]$对应4-bitx_{11}的分量。此时，计数器由100 bits变为44 bits。该部分的时间复杂度为$2^{72}\times(2^{100}\times 2^4\times 14)\approx 2^{179.81}$次4-bit S盒查询以及$2^{179.81}\times 5/11\approx 2^{178.67}$次4-bit线性计算。

(3)基于已猜测的128 bits密钥，猜测44-bit $\mathrm{EK}_{11}[i\sim(i+3)]$，$i\in$\{0, 4, 8, 12, 32, 68, 88, 92, 100, 120, 124\}。

针对每个4-bit猜测，部分解密z_{10}到x_{10}，并计算$z_9[0\sim3]$对应4-bitx_{10}的分量。此时计数器由44 bits变为4 bits。该部分所需要的时间复杂度为$2^{128}\times 2^{44}\times 2^8\times 11\approx 2^{183.46}$次4-bit S盒查询和$2^{183.6}/11\approx 2^{180}$次4-bit线性计算。

(4)基于已猜测的172 bits密钥，猜测4-bit $\mathrm{EK}_{10}[0\sim3]$。

对于每一个猜测，计算$x_9[3]$的对应值。保留让$x_9[3]=0$的密钥为正确密钥。该部分所需要的时间复杂度为$2^{172}\times 2^4\times 2^4=2^{180}$次4-bit S盒查询。

(5)穷搜剩余密钥。

每次猜测176-bit密钥后，256-bit的密钥空间都缩小1/2。因此只需要选取区分器中不同的平衡比

特位置，重复32次上述操作即可，最后遍历剩余的2^{256-32}密钥空间。

攻击复杂度：本文将字节替换和线性操作分别看作半轮加密，则

步骤(1)的时间复杂度为$2^{127}+2^{127}/2/12$次12轮加密；

步骤(2)，(3)和(4)的时间复杂度为T_1+T_2，其中　$T_1=(2^{140}+2^{151.81}+2^{164}+2^{179.81}+2^{183.46}+2^{180})/32$次128-bit S盒查询，$T_2=(2^{137.55}+2^{150.35}+2^{162.54}+2^{178.67}+2^{180})/32$次128-bit线性计算。

步骤(5)的时间复杂度为$32\times(T_1+T_2)/2/12+2^{224}\approx 2^{224}$次12轮加密。

综上所述，攻击12轮uBlock-128/128所需要的总时间复杂度为$T=2^{127}+2^{127}/2/12+2^{224}\approx 2^{224}$次12轮加密。内存复杂度为$M=(2^{136}\times 12)/8\approx 2^{138}$字节。数据复杂度为$2^{127}$选择明文。

5　结束语

本文对uBlock算法抵抗积分攻击的能力进行了进一步评估。首先，利用单项式传播技术结合MILP搜索工具搜索积分区分器。然后，基于搜索的积分区分器，利用部分和技术对不同版本的uBlock算法进行密钥恢复。对于uBlock-128/128，uBlock-128/256和uBlock-256/256，分别得到最长11/12/12轮的积分攻击。与之前的最优攻击相比，轮数分别提高2/3/2轮。本文的攻击说明，uBlock针对积分攻击依然有足够的安全冗余。

参 考 文 献

[1] BIHAM E and SHAMIR A. Differential cryptanalysis of DES-like cryptosystems[C]. Conference on the Theory and Application of Cryptography. Santa Barbara, USA, 1990: 2–21. doi: 10.1007/3-540-38424-3_1.

[2] BIHAM E and SHAMIR A. Differential cryptanalysis of DES-like cryptosystems[J]. *Journal of Cryptology*, 1991, 4(1): 3–72. doi: 10.1007/BF00630563.

[3] MATSUI M. Linear cryptanalysis method for DES cipher[C]. Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, 1993, 765: 386–397. doi: 10.1007/3-540-48285-7_33.

[4] DAEMEN J, KNUDSEN L, and RIJMEN V. The block cipher square[C]. The 4th International Workshop on Fast Software Encryption, Haifa, Israel, 1997: 149–165. doi: 10.1007/BFb0052343.

[5] KNUDSEN L and WAGNER D. Integral cryptanalysis[C]. The 9th International Workshop on Fast Software Encryption, Leuven, Belgium, 2002: 112–127. doi: 10.1007/3-540-45661-9_9.

[6] TODO Y. Structural evaluation by generalized integral property[C]. The 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, 2015: 287–314. doi: 10.1007/978-3-662-46800-5_12.

[7] TODO Y. Integral cryptanalysis on full MISTY1[C]. The 35th Annual Cryptology Conference, Santa Barbara, USA, 2015: 413–432. doi: 10.1007/978-3-662-47989-6_20.

[8] TODO Y. Integral cryptanalysis on full MISTY1[J]. *Journal of Cryptology*, 2017, 30(3): 920–959. doi: 10.1007/s00145-016-9240-x.

[9] TODO Y and MORII M. Bit-based division property and application to SIMON family[C]. The 23rd International Conference on Fast Software Encryption, Bochum, Germany, 2016: 357–377. doi: 10.1007/978-3-662-52993-5_18.

[10] XIANG Zejun, ZHANG Wentao, BAO Zhenzhen, *et al*. Applying MILP method to searching integral distinguishers based on division property for 6 lightweight block ciphers[C]. The 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, 2016: 648–678. doi: 10.1007/978-3-662-53887-6_24.

[11] DERBEZ P and LAMBIN B. Fast MILP models for division property[J]. *IACR Transactions on Symmetric Cryptology*, 2022, 2022(2): 289–321. doi: 10.46586/tosc.v2022.i2.289-321.

[12] ROHIT R and SARKAR S. Cryptanalysis of reduced round SPEEDY[C]. 13th International Conference on Cryptology in Africa, Fes, Morocco, 2022: 133–149. doi: 10.1007/978-3-031-17433-9_6.

[13] SHIBA R, SAKAMOTO K, LIU Fukang, *et al*. Integral and impossible-differential attacks on the reduced-round Lesamnta-LW-BC[J]. *IET Information Security*, 2022, 16(2): 75–85. doi: 10.1049/ise2.12044.

[14] SHIRAYA T, TAKEUCHI N, SAKAMOTO K, *et al*. MILP-based security evaluation for AEGIS/Tiaoxin-346/Rocca[J]. *IET Information Security*, 2023, 17(3): 458–467. doi: 10.1049/ise2.12109.

[15] WANG Senpeng, HU Bin, GUAN Jie, *et al*. MILP-aided method of searching division property using three subsets and applications[C]. The 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, 2019: 398–427. doi: 10.1007/978-3-030-34618-8_14.

[16] HAO Yonglin, LEANDER G, MEIER W, *et al*. Modeling for three-subset division property without unknown subset[C]. The 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques,

Zagreb, Croatia, 2020: 466–495. doi: 10.1007/978-3-030-45721-1_17.

[17] HAO Yonglin, LEANDER G, MEIER W, *et al*. Modeling for three-subset division property without unknown subset[J]. *Journal of Cryptology*, 2021, 34(3): 22. doi: 10.1007/s00145-021-09383-2.

[18] HU Kai, SUN Siwei, WANG Meiqin, *et al*. An algebraic formulation of the division property: Revisiting degree evaluations, cube attacks, and key-independent sums[C]. The 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, 2020: 446–476. doi: 10.1007/978-3-030-64837-4_15.

[19] CUI Jiamin, HU Kai, WANG Qingju, *et al*. Integral attacks on pyjamask-96 and round-reduced pyjamask-128[C]. Cryptographers' Track at the RSA Conference, Virtual Event, 2022: 223–246. doi: 10.1007/978-3-030-95312-6_10.

[20] CUI Jiamin, HU Kai, WANG Meiqin, *et al*. On the field-based division property: Applications to MiMC, feistel MiMC and GMiMC[C]. The 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, China, 2022: 241–270. doi: 10.1007/978-3-031-22969-5_9.

[21] 胡斌, 张贵显. μ^2算法的积分攻击和不可能差分攻击[J]. 电子与信息学报, 2022, 44(9): 3335–3342. doi: 10.11999/JEIT210638.
HU Bin and ZHANG Guixian. Integral cryptanalysis and impossible differential cryptanalysis of theμ^2algorithm[J]. *Journal of Electronics & Information Technology*, 2022, 44(9): 3335–3342. doi: 10.11999/JEIT210638.

[22] 吴文玲, 张蕾, 等. The Block Cipher uBlock [OL]. https://sfjs.cacrnet.org.cn/site/term/list_76_1.html.

[23] 吴文玲, 张蕾, 郑雅菲, 等. 分组密码uBlock[J]. 密码学报, 2019, 6(6): 690–703. doi: 10.13868/j.cnki.jcr.000334.
WU Wenling, ZHANG Lei, ZHENG Yafei, *et al*. The block cipher uBlock[J]. *Journal of Cryptologic Research*, 2019, 6(6): 690–703. doi: 10.13868/j.cnki.jcr.000334.

[24] TIAN Wenqiang and HU Bin. Integral cryptanalysis on two block ciphers pyjamask and uBlock[J]. *IET Information Security*, 2020, 14(5): 572–579. doi: 10.1049/iet-ifs.2019.0624.

[25] MAO Yongxia, WU Wenling, WANG Bolin, *et al*. Improved division property for ciphers with complex linear layers[C]. The 27th Australasian Conference on Information Security and Privacy, Wollongong, Australia, 2022: 106–124. doi: 10.1007/978-3-031-22301-3_6.

[26] 黄明, 张莎莎, 洪春雷, 等. 分组密码复杂线性层可分性传播的MILP刻画方法[J]. 软件学报, 2023: 1–13. doi: 10.13328/j.cnki.jos.006839.
HUANG Ming, ZHANG Shasha, HONG Chunlei, *et al*. MILP modeling of division property propagation for block ciphers with complex linear layers[J]. *Journal of Software*, 2023: 1–13. doi: 10.13328/j.cnki.jos.006839.

[27] https://www.sagemath.org.

[28] SUN Siwei, HU Lei, WANG Peng, *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]. The 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, China, 2014: 158–178. doi: 10.1007/978-3-662-45611-8_9.

[29] SUN Ling, WANG Wei, and WANG Meiqin. MILP-aided bit-based division property for primitives with non-bit-permutation linear layers[J]. *IET Information Security*, 2020, 14(1): 12–20. doi: 10.1049/iet-ifs.2018.5283.

[30] FERGUSON N, KELSEY J, LUCKS S, *et al*. Improved cryptanalysis of rijndael[C]. The 7th International Workshop on Fast Software Encryption, New York, USA, 2000: 213–230. doi: 10.1007/3-540-44706-7_15.

王 晨：女，博士生，研究方向为分组密码算法的安全性分析.
崔佳敏：女，博士生，研究方向为对称密码算法的安全性分析.
李木舟：男，博士，研究方向为对称密码算法的分析与设计.
王美琴：女，教授，研究方向为对称密码算法的分析与设计.

责任编辑：马秀强