计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.20220485
59(10):2275-2285,2022

uBlock 类结构最优向量置换的高效搜索

李晓丹[1,2,3]　吴文玲[1,2]　张　丽[1,2]
[1](中国科学院软件研究所可信计算与信息保障实验室　北京　100190)
[2](中国科学院大学　北京　100049)
[3](中国星网网络系统研究院有限公司　北京　100083)
(xiaodan2018@iscas.ac.cn)

Efficient Search for Optimal Vector Permutations of uBlock-like Structures

Li Xiaodan[1,2,3], Wu Wenling[1,2], and Zhang Li[1,2]
[1](*Trusted Computing and Information Assurance Laboratory*, *Institute of Software*, *Chinese Academy of Sciences*, *Beijing* 100190)
[2](*University of Chinese Academy of Sciences*, *Beijing* 100049)
[3](*China Satellite Network System Institute Co.*, *Ltd.*, *Beijing* 100083)

Abstract　The overall structure is an important feature of block cipher and also the primary research object. It has a great influence on the performance of hardware and software in the selection of rounds of block cipher. In the design process of the AES-like ciphers, when using a matrix with a non-optimal branch number for the MixColumns operation, the choice of the vector permutation, i.e., an alternative for ShiftRows, can actually improve the security of the primitive. uBlock-like structure is an AES-like structure. In this paper, we investigate the characteristics and diffusivity of uBlock-like structures, the lower bound of the number of full diffusion rounds and the equivalence class division criteria, and then we propose a search strategy for optimal vector permutations of uBlock-like structures. According to the optimal number of full diffusion rounds, the optimal branch number of the super diffusion layer, and the special properties of the diffusion layer of uBlock-like structure, we prove that the left and right vector permutations cannot be the identity transformation, and a series of optimal vector permutations of uBlock-like structures are given. The search strategy greatly reduces the number of permutation pairs that need to be tested and provides technical support for the design of uBlock-like algorithms.

Key words　block cipher; uBlock-like structures; full diffusion; super S-box; optimal permutation

摘　要　整体结构是分组密码的重要特征,也是首要的研究对象,对于分组密码的轮数选取、软硬件实现性能都有非常大的影响.对于类 AES 算法的设计,当选用非最优分支数的矩阵作为列混淆操作时,向量置换(即字换位操作)的选择可有效提高整体结构的安全性.uBlock 类结构是一种类 AES 结构,通过研究 uBlock 类结构的特点及其扩散性,给出了其全扩散轮数的下界及等价类划分准则,提出了一种 uBlock 类结构最优向量置换的搜索策略.依据全扩散轮数最优、超级扩散层的分支数最优及 uBlock 类结构扩散层的特殊性质,证明了左右向量置换都不能是恒等变换,给出了 uBlock 类结构的一系列最优向量置换.该搜索策略大幅度减少了需要测试的置换对,为后续 uBlock 类算法的设计提供技术支持.

收稿日期:2022-06-10;**修回日期**:2022-08-09
基金项目:国家自然科学基金项目(62072445)
This work was supported by the National Natural Science Foundation of China (62072445).

关键词 分组密码;uBlock 类结构;全扩散;超级 S 盒;最优置换

中图法分类号 TP309

在分组密码的设计和分析中,整体结构是首要的研究对象.作为分组密码的重要特征,整体结构对于分组密码的轮数选取、软硬件实现性能都有非常大的影响.常用的分组密码整体结构有:Feistel 结构、SP 结构、广义 Feistel 结构、MISTY 结构、Lai-Massey 结构以及由上述结构彼此嵌套形成的细化整体结构.其中,SP 结构非常清晰,是直接基于香农的"混淆"和"扩散"原则实现的整体结构,通常包含一个可逆的非线性函数 S 和一个可逆线性变换 P,其中 S 层起混淆作用,线性层 P 起扩散作用.当给定 S 层和 P 层的某些密码指标,设计者可给出 SP 密码抵抗差分分析和线性分析的可证明安全.相较于 Feistel 结构,SP 结构扩散速度更快,但是为达到加密和解密的相似性需要合理设计密码部件.SP 结构的分组密码算法有:AES[1],ARIA[2] 和 Serpent 等.其中 AES 无疑是目前最重要的分组密码,它的扩散层由 2 部分组成,也被称为两级扩散层,一部分是选取分支数最大的 MDS 矩阵作为列混淆,另一部分是行移位操作.特别地,许多分组密码和杂凑函数的设计都是从 AES 的初始设计结构开始,对一个或多个部件进行调整以满足其设计要求,如 Anubis[3],LED[4],Midori[5],PHOTON[6],QARMA[7],SKINNY[8] 和 Whirlpool[9].此类算法的线性层本身由 2 部分组成:一部分类似于 AES 的列混合操作,另一部分类似于 AES 行移位操作,我们称这类密码算法为类 AES 密码算法.列混合操作是对状态列的矩阵乘法,行移位操作是对状态字的置换.对于前者,研究结果较多,只需要保证其具有高的分支数,分支数越高,对应的活跃 S 盒数越多,则抵抗差分分析和线性分析的能力越强.AES 选取的列混合操作为分支数最大的 MDS 矩阵,而出于轻量化考虑的一些算法,如 Midori,SKINNY 等,则采用非最优分支数的二元矩阵.而对于字换位操作,当仅考虑超过 2 轮的情形时,字换位的选取极大影响活跃 S 盒的数量,因此,对于好的设计来说,精心选择字换位操作至关重要.对于 AES,得益于宽轨迹设计策略[10],可以获得数学上可证明的最小活跃 S 盒数的界限,保证 4 轮后至少有 25 个活跃 S 盒.Midori 算法设计的初衷是减少硬件资源损耗,它采用类似于 AES 算法的结构,与 SKINNY 算法类似,它们都属于类 AES 算法,但是在列混淆部分选用非最优分支数的矩阵.Midori 设计者发现使用分支数为 4 的二元矩阵时,4 轮后活跃 S 盒数下降到 16 个,但改变字换位操作可显著提高活跃 S 盒数.2015 年,文献[11]证明了对于选用最优分支数的列混合操作的算法,用任意置换替换字换位操作不能增加活跃 S 盒的数量.2016 年,文献[12]给出,用 B 表示矩阵的分支数,对于经典字换位操作,则对于列混合为 MDS 矩阵的,4 轮后至少有 B^2 个活跃 S 盒;对于列混合为二元矩阵的,4 轮后活跃 S 盒的下界可能大于 B^2,且对于某些二元矩阵,下界可达到 $B(B+2)$.2018 年,文献[13]提出一种加速搜索字换位操作的技术,并应用于 Midori 和 SKINNY 算法,寻找使得整体扩散性更好的字换位操作.这也是现在轻量级分组密码的一个设计趋势,扩散层选择非 MDS 矩阵,虽然扩散效果没有 MDS 矩阵好,但是实现效率会有很大提高,这在资源受限的环境下有很大优势,而且在结合合适的向量置换操作后,整体算法也可以达到较好的扩散性和安全性.因此,使用类 AES 结构来设计轻量级分组密码是一个不错的选择,特别是列混合操作选用最优二元扩散层,并结合恰当的向量置换,可以很好地平衡安全性和实现代价,然而在选定列混合操作后,搜索合适的向量置换并不容易,这也是这类算法在设计时需要花费大量精力的部分.

uBlock 算法[14] 是全国密码算法设计竞赛中的获胜算法.该算法是经典的 SP 结构,是一种典型的类 AES 算法,线性层选用的是 16 维的最优二元扩散层与向量置换的组合,扩散速度快且可以在各种软硬件平台上高速实现.受文献[13]的启发,本文研究了 uBlock 类结构的扩散性,并给出了寻找最优置换的搜索策略.通过对 uBlock 类结构中二元扩散层性质的研究,我们给出 uBlock 类结构全扩散轮数的下界.根据结构特点,我们揭示了 uBlock 类结构等价类的划分准则,并基于此给出了 uBlock 类结构最优向量置换的搜索策略.最后根据 128 b 和 256 b 分组的 uBlock 类结构的特点,进一步优化了搜索策略,并依据全扩散轮数、性能和超级扩散层的分支数 3 个指标,给出了 128 b 和 256 b 分组的 uBlock 类结构的一系列最优向量置换.我们的方法可以大幅度降低需要测试的置换对,为后续 uBlock 类算法的设计提供技术支持.

1 基础知识

本节介绍相关的基础概念和定义.

1.1 符号表示

表 1 给出了本文中使用的符号:

Table 1 Symbols

表 1 符号表示

符号	含义
F_2	二元有限域
F_2^m	F_2 上的 m 维向量空间
$\{0,1\}^n$	长度为 n 的比特串的集合
$\boldsymbol{M}^{\mathrm{T}}$	矩阵 $\boldsymbol{M}$ 的转置
$\boldsymbol{Circ}(*)$	首行为 $*$ 的循环矩阵

1.2 分支数

定义 1. 给定一个二元向量 $\boldsymbol{x}\in F_2^{nm}$,可看作是 n 个 m 比特串的连接.F_2^m 上 $\boldsymbol{x}$ 的汉明重量定义为 $\boldsymbol{x}$ 的非零 m 比特串的个数,表示为 $wt(\boldsymbol{x})$.

定义 2[10]. F_2^m 上的矩阵 $\boldsymbol{M}$ 的差分分支数定义为

$$B_{\mathrm{diff}}(\boldsymbol{M})=\min_{\boldsymbol{x}\in F_2^{nm}\setminus\{0\}}\{wt(\boldsymbol{x})+wt(\boldsymbol{Mx})\}.$$

$\boldsymbol{M}$ 的线性分支数定义为

$$B_{\mathrm{linear}}(\boldsymbol{M})=\min_{\boldsymbol{x}\in F_2^{nm}\setminus\{0\}}\{wt(\boldsymbol{x})+wt(\boldsymbol{M}^{\mathrm{T}}\boldsymbol{x})\}.$$

分支数反映了密码方案的扩散性,此外也与差分分析和线性分析相关.分支数达到最大的矩阵称为 MDS 矩阵,分支数达到次优的矩阵称为 Near-MDS 矩阵.基于分支数,密码方案的设计者和分析者可以估计活跃 S 盒的下界,从而评估算法抵抗差分分析和线性分析的能力.

1.3 uBlock 算法描述

uBlock 算法的整体结构采用 PX(Pshufb-Xor)结构(SP 结构的一种细化结构),Pshufb 和 Xor 分别是向量置换和异或运算指令.算法设计采用 S 盒和分支数的理念,对差分分析和线性分析具有可证明的安全性,同时对于不可能差分分析、积分分析、中间相遇攻击等分析方法具有相对成熟的分析评估理论支持.此外,uBlock 算法适应各种软硬件平台,充分考虑了微处理器的计算资源,可以利用 SSE,AVX2 和 NEON 等指令集高效实现;硬件实现简单而有效,既可以高速实现,满足高性能环境的应用需求,也可以轻量化实现,满足资源受限环境的安全需求.

uBlock 是一族分组密码算法,分组长度和密钥长度支持 128 b 和 256 b,分别记为 uBlock128/128,uBlock128/256 和 uBlock256/256,迭代轮数分别为 16,24 和 24.加密算法由轮迭代变换组成,轮变换如图 1 所示:

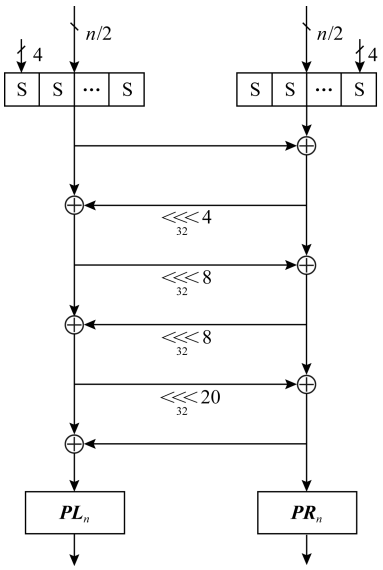


Fig. 1 Round function of uBlock algorithm

图 1 uBlock 算法轮函数

图 1 中,基本模块为:

1) S 盒 $\boldsymbol{S}_n$.$\boldsymbol{S}_n$ 是由 $\frac{n}{8}$ 个相同的 4 b 的 S 盒并置而成,定义为

$$\boldsymbol{S}_n:(\{0,1\}^4)^{\frac{n}{8}}\rightarrow(\{0,1\}^4)^{\frac{n}{8}},$$

$$(x_0,x_1,\cdots,x_{\frac{n}{8}-1})\rightarrow(\boldsymbol{S}(x_0),\boldsymbol{S}(x_1),\cdots,\boldsymbol{S}(x_{\frac{n}{8}-1})).$$

4 b 的 S 盒如表 2 所示:

Table 2 The 4 bit S Box of uBlock Algorithm

表 2 uBlock 算法 4 b 的 S 盒

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

2) $\boldsymbol{PL}_n$ 和 $\boldsymbol{PR}_n$.$\boldsymbol{PL}_n$ 和 $\boldsymbol{PR}_n$ 都是 $m\left(m=\frac{n}{16}\right)$ 个字节的向量置换,如表 3 所示:

Table 3 $\boldsymbol{PL}_n$ and $\boldsymbol{PR}_n$

表 3 $\boldsymbol{PL}_n$ 和 $\boldsymbol{PR}_n$

$\boldsymbol{PL}_n$ 和 $\boldsymbol{PR}_n$	向量置换
$\boldsymbol{PL}_{128}$	(1,3,4,2,0,6,7,5)
$\boldsymbol{PR}_{128}$	(2,7,5,0,1,6,4,3)
$\boldsymbol{PL}_{256}$	(2,7,8,13,3,6,9,12,1,4,15,10,14,11,5,0)
$\boldsymbol{PR}_{256}$	(6,11,1,12,9,4,2,15,7,0,13,10,14,3,8,5)

比如 $\boldsymbol{PL}_{128}$ 表示为

$$\boldsymbol{PL}_{128}:(\{0,1\}^8)^8\to(\{0,1\}^8)^8$$
$$(y_0,y_1,\cdots,y_7)\to(z_0,z_1,\cdots,z_7)$$
$$z_0=y_1,z_1=y_3,z_2=y_4,z_3=y_6,$$
$$z_4=y_0,z_5=y_2,z_6=y_7,z_7=y_5.$$

2 uBlock 类结构

uBlock 算法采用的整体结构是 PX 结构，其扩散层由 2 部分组成：一部分是线性变换，即由 Feistel 结构构造的二元最优扩散层；另一部分是向量置换. 在本节中，我们探索一类 PX 结构的扩散特性，即线性变换部分选取与 uBlock 算法相同的二元扩散层，向量置换部分与 uBlock 算法不同的结构，称之为 uBlock 类结构，形式如图 2 所示：

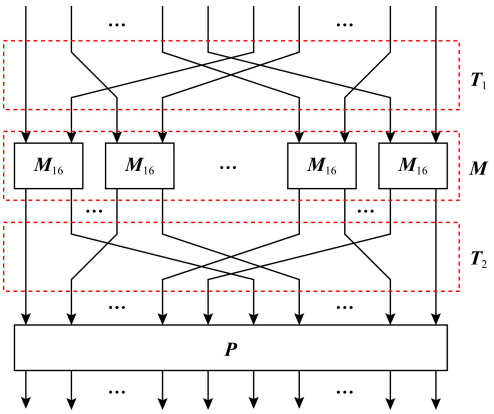


Fig. 2 Diffusion layer of uBlock-like structure

图 2 uBlock 类结构的扩散层

首先，我们给出 uBlock 类结构的矩阵描述，其可以表示为 $\boldsymbol{M}_n=\boldsymbol{P}\circ\boldsymbol{T}_2\circ\boldsymbol{M}\circ\boldsymbol{T}_1$，其中 n 表示分组长度，$\boldsymbol{T}_1,\boldsymbol{T}_2$ 为 $\frac{n}{32}$ 个元素的置换，$\boldsymbol{M}$ 为 $\frac{n}{64}$ 个 $\boldsymbol{M}_{16}$ 的并置，$\boldsymbol{P}$ 为向量置换.

此外，我们观察到 $\boldsymbol{M}_{16}$ 用矩阵形式表示出来，恰好为如下形式的分块矩阵：

$$\boldsymbol{M}_{16}=\begin{pmatrix}\boldsymbol{A} & \boldsymbol{B}\\ \boldsymbol{B} & \boldsymbol{C}\end{pmatrix},$$

其中，

$$\boldsymbol{A}=\boldsymbol{Circ}(0,0,1,1,1,0,1,1),$$
$$\boldsymbol{B}=\boldsymbol{Circ}(1,1,0,1,1,0,1,1),$$
$$\boldsymbol{C}=\boldsymbol{Circ}(1,0,1,1,0,0,1,1).$$

由于

$$\boldsymbol{T}_1=\left(1,\frac{n}{64}+1,2,\frac{n}{64}+2,3,\frac{n}{64}+3,\cdots,\frac{n}{64},\frac{n}{32}\right),$$

$$\boldsymbol{T}_2=\left(1,3,5,\cdots,\frac{n}{32}-1,2,4,\cdots,\frac{n}{32}\right),$$

因此

$$\boldsymbol{T}_2\cdot\boldsymbol{M}\cdot\boldsymbol{T}_1=\begin{pmatrix}\boldsymbol{A}_m & \boldsymbol{B}_m\\ \boldsymbol{B}_m & \boldsymbol{C}_m\end{pmatrix},$$

其中 $\boldsymbol{A}_m,\boldsymbol{B}_m,\boldsymbol{C}_m$ 指对角线元素分别为 $\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}$，而其余元素为零矩阵的 $m\times m$ 分块矩阵，即有

$$\boldsymbol{M}_n=\boldsymbol{P}\cdot\boldsymbol{T}_2\cdot\boldsymbol{M}\cdot\boldsymbol{T}_1=\boldsymbol{P}\cdot\begin{pmatrix}\boldsymbol{A}_m & \boldsymbol{B}_m\\ \boldsymbol{B}_m & \boldsymbol{C}_m\end{pmatrix}.$$

我们的目标是探索 uBlock 类结构的扩散性，并寻找最优的向量置换使得整体结构的扩散性和安全性达到最优，即我们需要考虑的指标有 3 个：

1) 全扩散轮数. 全扩散轮数越小，算法的扩散性越强，且可以根据全扩散轮数大致估计算法抵抗不可能差分、积分分析等结构性分析方法的能力. 因此，我们旨在寻找可以达到最小全扩散轮数的置换.

2) 实现性能. 尽可能减少软件实现中的指令数.

3) 4 轮超级 S 盒下的分支数. 我们可以根据分支数的概念给出算法活跃 S 盒的下界，并基于此估计其抵抗差分和线性分析的能力. 因此，我们希望分支数越大越好.

2.1 全扩散轮数

扩散最初是由香农在文献[15]中定义的，意思是一个子块的输入影响全部子块的输出. 文献[16]证明了抵抗饱和攻击和不可能差分攻击的轮数与称之为全扩散轮数的概念相关，用 DR 表示全扩散轮数. 同时，证明了给定的结构需要至少 $2DR+1$ 轮来抵抗上述攻击. 因此，在设计分组密码时，全扩散轮数是一个重要的参考指标. 接下来，我们研究 uBlock 类结构全扩散轮数的下界，首先给出 $\boldsymbol{M}_{16}$ 的扩散性质：

性质 1. 输入向量的汉明重量为 1 时，经过 $\boldsymbol{M}_{16}$ 均可扩散到 11 个位置.

性质 2. 输入向量的汉明重量为 2 时，有如下 8 种情形在经过 $\boldsymbol{M}_{16}$ 后可全扩散：

$$(1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0),$$
$$(0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1),$$
$$(0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0),$$
$$(0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0),$$
$$(0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0),$$
$$(0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0),$$
$$(0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0),$$
$$(0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0).$$

性质 3. 输入向量的汉明重量为 3 时，有 400 种

输入在经过 $\boldsymbol{M}_{16}$ 后可全扩散,剩余 160 种输入均可扩散到 15 个位置.

性质 4. 输入向量的汉明重量为 4 时,有 1 740 种输入在经过 $\boldsymbol{M}_{16}$ 后可全扩散,剩余 80 种输入均可扩散到 15 个位置.

性质 5. 输入向量的汉明重量为 5 时,有 4 352 种输入在经过 $\boldsymbol{M}_{16}$ 后可全扩散,剩余 16 种输入均可扩散到 15 个位置.

性质 6. 输入向量的汉明重量大于 5 时,经过 $\boldsymbol{M}_{16}$ 后均可全扩散.

根据上述 6 个性质可知,要使经过 $\boldsymbol{M}_{16}$ 后全扩散,输入向量的汉明重量至少为 2.则对于 uBlock 类结构,假定输入向量的汉明重量为 1,则经过 1 轮轮函数后可扩散到 11 个位置,这 11 个位置经过 2 轮至多可扩散到 $16\times5+11=91$ 个位置,经过 3 轮至多可扩散到 $45\times16+11=731$ 个位置.因此,我们可以给出一系列 uBlock 类结构的全扩散轮数下界,如表 4 所示.表 4 显示了 uBlock 类结构具有极强的扩散性.

Table 4 Lower Bound of Full Diffusion Round for uBlock-like Structures

表 4 uBlock 类结构全扩散轮数的下界

$\boldsymbol{M}_{16}$ 的个数	分组规模	全扩散轮数下界
2	128	2
4	256	2
6	384	3
8	512	3
16	1 024	3

2.2 超级 S 盒

为了进一步估计算法抵抗差分和线性分析的能力,我们引入超级 S 盒的概念.uBlock 类结构连续 4 轮轮函数作用在中间状态 $\boldsymbol{X}$ 上可以表示为

$$\boldsymbol{P}\circ\boldsymbol{T}_2\circ\boldsymbol{M}\circ\boldsymbol{T}_1\circ\boldsymbol{S}\circ\boldsymbol{P}\circ\boldsymbol{T}_2\circ\boldsymbol{M}\circ\boldsymbol{T}_1\circ\boldsymbol{S}\circ\boldsymbol{P}\circ\boldsymbol{T}_2\circ\boldsymbol{M}\circ\boldsymbol{T}_1\circ\boldsymbol{S}\circ\boldsymbol{P}\circ\boldsymbol{T}_2\circ\boldsymbol{M}\circ\boldsymbol{T}_1\circ\boldsymbol{S}(\boldsymbol{X}). \tag{1}$$

注意到 $\boldsymbol{S}$ 和 $\boldsymbol{T}_1,\boldsymbol{T}_2,\boldsymbol{P}$ 可以交换位置,因此式(1)等价于

$$\boldsymbol{P}\circ\boldsymbol{T}_2\circ\boldsymbol{S}\circ\boldsymbol{M}\circ\boldsymbol{S}\circ\boldsymbol{T}_1\circ\boldsymbol{P}\circ\boldsymbol{T}_2\circ\boldsymbol{M}\circ\boldsymbol{T}_1\circ\boldsymbol{P}\circ\boldsymbol{T}_2\circ\boldsymbol{S}\circ\boldsymbol{M}\circ\boldsymbol{S}\circ\boldsymbol{T}_1\circ\boldsymbol{P}\circ\boldsymbol{T}_2\circ\boldsymbol{M}\circ\boldsymbol{T}_1(\boldsymbol{X}),$$

其中 $\boldsymbol{S}\circ\boldsymbol{M}\circ\boldsymbol{S}$ 可以看作 $\frac{n}{64}$ 个 16×16 的超级 S 盒 $S_{\text{super}}=\boldsymbol{S}\circ\boldsymbol{M}_{16}\circ\boldsymbol{S}$ 的并置.由于 $\boldsymbol{M}_{16}$ 的分支数为 8,因此每一个活跃的超级 S 盒至少有 8 个活跃的 4 b 的 S 盒.

相应的 $\boldsymbol{M}_{\text{super}}=\boldsymbol{T}_1\circ\boldsymbol{P}\circ\boldsymbol{T}_2\circ\boldsymbol{M}\circ\boldsymbol{T}_1\circ\boldsymbol{P}\circ\boldsymbol{T}_2$ 为超级线性层.因此,可以通过 $\boldsymbol{M}_{\text{super}}$ 的分支数来估计算法的活跃 S 盒数,进而估计其抵抗差分和线性分析的能力,即

$$\boldsymbol{M}_{\text{super}}=\boldsymbol{T}_1\cdot\boldsymbol{P}\cdot\boldsymbol{T}_2\cdot\boldsymbol{M}\cdot\boldsymbol{T}_1\cdot\boldsymbol{P}\cdot\boldsymbol{T}_2=\boldsymbol{T}_1\cdot\boldsymbol{P}\cdot\begin{pmatrix}\boldsymbol{A}_m & \boldsymbol{B}_m\\ \boldsymbol{B}_m & \boldsymbol{C}_m\end{pmatrix}\cdot\boldsymbol{P}\cdot\boldsymbol{T}_2, \tag{2}$$

因此,我们可以直接通过式(2)来计算 $\boldsymbol{M}_{\text{super}}$ 的分支数.

2.3 等价类划分准则

为了减少搜索空间,我们进一步探索 uBlock 类结构的等价类划分准则,并给出最优向量置换的搜索策略.首先,uBlock 类结构可用图 3 描述:

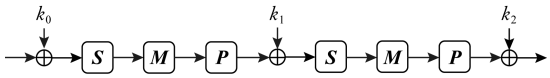


Fig. 3 uBlock-like structures

图 3 uBlock 类结构

显然,直接搜索全部的置换 $\boldsymbol{P}$ 是不现实的,即使在 uBlock 算法中,$\boldsymbol{P}$ 指 $\boldsymbol{PL}$ 和 $\boldsymbol{PR}$ 的并置,此时全部置换的搜索复杂度为 $\left(\frac{n}{16}!\right)^2$,这在分组规模变大时搜索工作也是极其困难的.为了减少搜索空间,我们希望给所有置换做等价类划分,即给出一些条件使得在同一等价类中的 2 个置换具有相同的密码学性质.特别地,与文献[13]中的结构类似,我们观察到对置换 $\boldsymbol{Q}$,假若 $\boldsymbol{MQ}=\boldsymbol{QM}$,则 $\boldsymbol{P}$ 与 $\boldsymbol{QPQ}^{-1}$ 在 uBlock 类结构中具有相同的密码学性质.接下来,我们用图示给出证明.

显然,图 4(a)和 4(b)两种形式是等价的,此时假若 $\boldsymbol{MQ}=\boldsymbol{QM}$,则图 4(c)与图 4(a)和 4(b)也是等价的,即可得到图 4(d)与 4(a)等价,即假若 $\boldsymbol{MQ}=\boldsymbol{QM}$,则 $\boldsymbol{P}$ 与 $\boldsymbol{QPQ}^{-1}$ 在 uBlock 类结构中具有相同的密码学性质.

于是,我们可以给出如下搜索策略:

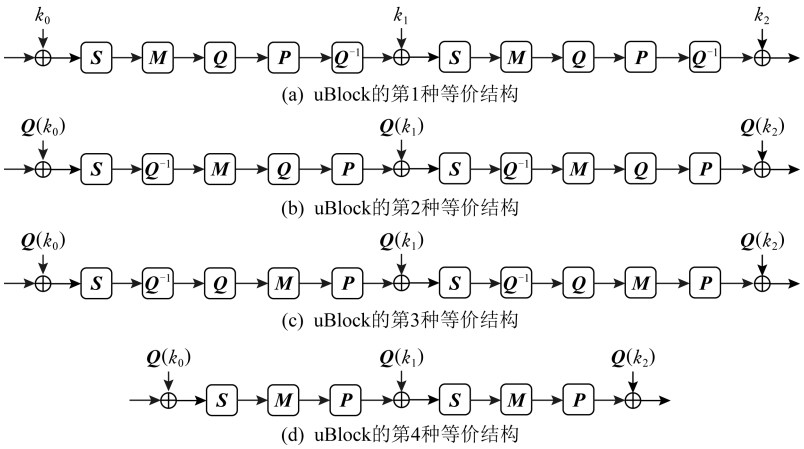
策略 1. uBlock 类结构最优向量置换的搜索策略.

步骤 1. 确定使得 $\boldsymbol{MQ}=\boldsymbol{QM}$ 的所有置换 $\boldsymbol{Q}$;

步骤 2. 对得到的所有置换 $\boldsymbol{Q}$,则 $\boldsymbol{P}$ 与 $\boldsymbol{QPQ}^{-1}$ 属于同一个等价类.

步骤 3. 对每一个等价类中的代表元,测试算法整体的全扩散轮数.

步骤 4. 对全扩散轮数最小的置换,检测 $\boldsymbol{M}_{\text{super}}$ 及其逆变换的分支数.



(a) uBlock的第1种等价结构

(b) uBlock的第2种等价结构

(c) uBlock的第3种等价结构

(d) uBlock的第4种等价结构

Fig. 4 Four equivalent structures of the uBlock-like structure

图 4 uBlock 类结构的 4 种等价结构

3 应 用

我们将第 2 节的搜索策略应用于 128 b 分组和 256 b 分组的 uBlock 类结构中，寻找最优向量置换. 对于 128 b 和 256 b 分组的 uBlock 类结构，我们选择与 uBlock 算法中的结构相同，即 $\boldsymbol{P}$ 层是 $\boldsymbol{PL}$ 和 $\boldsymbol{PR}$ 的并置，且均为面向字节的向量置换，因此我们可以进一步优化搜索策略 1.

3.1 128 b 分组

128 b 分组的 uBlock 类结构的扩散层描述如图 5 所示：

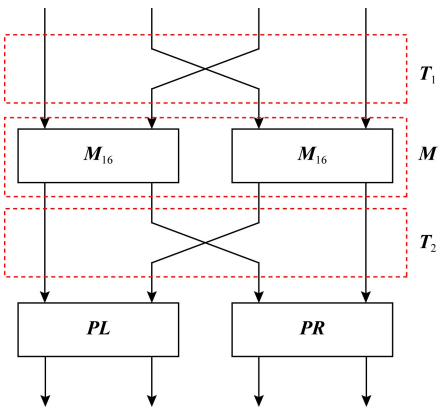


Fig. 5 Diffusion layer of uBlock-like structure with 128 bit sizes

图 5 128 b 分组的 uBlock 类结构的扩散层

其中

$$\boldsymbol{T}_1=\boldsymbol{T}_2=\begin{pmatrix}\boldsymbol{I}&\boldsymbol{O}&\boldsymbol{O}&\boldsymbol{O}\\\boldsymbol{O}&\boldsymbol{O}&\boldsymbol{I}&\boldsymbol{O}\\\boldsymbol{O}&\boldsymbol{I}&\boldsymbol{O}&\boldsymbol{O}\\\boldsymbol{O}&\boldsymbol{O}&\boldsymbol{O}&\boldsymbol{I}\end{pmatrix},$$

则

$$\boldsymbol{M}_{128}=\begin{pmatrix}\boldsymbol{PL}&\boldsymbol{O}\\\boldsymbol{O}&\boldsymbol{PR}\end{pmatrix}\begin{pmatrix}\boldsymbol{A}_2&\boldsymbol{B}_2\\\boldsymbol{B}_2&\boldsymbol{C}_2\end{pmatrix}=\begin{pmatrix}\boldsymbol{PLA}_2&\boldsymbol{PLB}_2\\\boldsymbol{PRB}_2&\boldsymbol{PRC}_2\end{pmatrix}.$$

其中，$\boldsymbol{I}$ 为单位矩阵，$\boldsymbol{O}$ 为零矩阵.

由表 3 可知，128 b 分组的 uBlock 类结构至少需要 2 轮全扩散. 因此，我们仅需要寻找 2 轮全扩散下的最优向量置换. 出于性能的考虑，若 $\boldsymbol{PL}$ 或 $\boldsymbol{PR}$ 是恒等变换，则轮函数少一个指令，此时算法软件性能会有所提升. 使用等价类划分技术，我们发现 $\boldsymbol{PL}$ 和 $\boldsymbol{PR}$ 是恒等变换时均有 27 个等价类满足 2 轮全扩散. 部分具体实例在附录中给出.

此外，我们考虑 4 轮超级 S 盒下，扩散层的分支数达到最优的情形. 由于

$$\boldsymbol{M}_{\text{super}}=\begin{pmatrix}\boldsymbol{I}&\boldsymbol{O}&\boldsymbol{O}&\boldsymbol{O}\\\boldsymbol{O}&\boldsymbol{O}&\boldsymbol{I}&\boldsymbol{O}\\\boldsymbol{O}&\boldsymbol{I}&\boldsymbol{O}&\boldsymbol{O}\\\boldsymbol{O}&\boldsymbol{O}&\boldsymbol{O}&\boldsymbol{I}\end{pmatrix}\cdot$$

$$\begin{pmatrix}\boldsymbol{PLA}_2\boldsymbol{PL}&\boldsymbol{PLB}_2\boldsymbol{PR}\\\boldsymbol{PRB}_2\boldsymbol{PL}&\boldsymbol{PRC}_2\boldsymbol{PR}\end{pmatrix}\begin{pmatrix}\boldsymbol{I}&\boldsymbol{O}&\boldsymbol{O}&\boldsymbol{O}\\\boldsymbol{O}&\boldsymbol{O}&\boldsymbol{I}&\boldsymbol{O}\\\boldsymbol{O}&\boldsymbol{I}&\boldsymbol{O}&\boldsymbol{O}\\\boldsymbol{O}&\boldsymbol{O}&\boldsymbol{O}&\boldsymbol{I}\end{pmatrix},$$

我们可将扩散层看作一个 2×2 的分块矩阵，考虑其为 MDS 矩阵，即分支数为 3. 此时，根据超级 S 盒和分支数的概念，可知其 4 轮至少有 24 个活跃 S 盒（与 uBlock-128 中选择的向量置换在 4 轮时的活跃 S 盒数一致）.

由于 uBlock 算法中 $\boldsymbol{P}$ 的形式为 $\begin{pmatrix}\boldsymbol{PL}&\boldsymbol{O}\\\boldsymbol{O}&\boldsymbol{PR}\end{pmatrix}$，所以我们考虑 $\boldsymbol{Q}$ 的形式为 $\begin{pmatrix}\boldsymbol{Q}_1&\boldsymbol{O}\\\boldsymbol{O}&\boldsymbol{Q}_2\end{pmatrix}$. 则有结论：

假设 $\boldsymbol{Q}=\begin{pmatrix}\boldsymbol{Q}_1 & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{Q}_2\end{pmatrix}$,要使 $\boldsymbol{QM}=\boldsymbol{MQ}$,则 $\boldsymbol{Q}$ 需满足条件:

$$\boldsymbol{Q}_1\boldsymbol{A}_2=\boldsymbol{A}_2\boldsymbol{Q}_1,\quad \boldsymbol{Q}_2\boldsymbol{C}_2=\boldsymbol{C}_2\boldsymbol{Q}_2,$$
$$\boldsymbol{Q}_1\boldsymbol{B}_2=\boldsymbol{B}_2\boldsymbol{Q}_2,\quad \boldsymbol{Q}_2\boldsymbol{B}_2=\boldsymbol{B}_2\boldsymbol{Q}_1.$$

策略 2. uBlock-128 最优向量置换的搜索策略.

步骤 1. 寻找使得 $\boldsymbol{Q}_3\cdot\boldsymbol{A}_2=\boldsymbol{A}_2\cdot\boldsymbol{Q}_3$ 的所有置换 $\boldsymbol{Q}_3$,记为集合 $S_{\boldsymbol{Q}_3}$;

步骤 2. 寻找使得 $\boldsymbol{Q}_4\cdot\boldsymbol{C}_2=\boldsymbol{C}_2\cdot\boldsymbol{Q}_4$ 的所有置换 $\boldsymbol{Q}_4$,记为集合 $S_{\boldsymbol{Q}_4}$;

步骤 3. 对每个 $\boldsymbol{Q}_1\in S_{\boldsymbol{Q}_3}$,$\boldsymbol{Q}_2\in S_{\boldsymbol{Q}_4}$,寻找满足 $\begin{cases}\boldsymbol{Q}_1\boldsymbol{B}_2=\boldsymbol{B}_2\boldsymbol{Q}_2\\ \boldsymbol{Q}_2\boldsymbol{B}_2=\boldsymbol{B}_2\boldsymbol{Q}_1\end{cases}$ 的集合对 $S_{\boldsymbol{Q}_1,\boldsymbol{Q}_2}$;

步骤 4. 使用集合对 $S_{\boldsymbol{Q}_1,\boldsymbol{Q}_2}$ 中的置换,对 $\boldsymbol{P}$ 的全空间做等价类划分,若

$$\begin{pmatrix}\boldsymbol{Q}_1 & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{Q}_2\end{pmatrix}\begin{pmatrix}\boldsymbol{P}_1 & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{P}_2\end{pmatrix}\begin{pmatrix}\boldsymbol{Q}_1^{-1} & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{Q}_2^{-1}\end{pmatrix}=\begin{pmatrix}\boldsymbol{P}_3 & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{P}_4\end{pmatrix},$$

则

$$\begin{pmatrix}\boldsymbol{P}_1 & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{P}_2\end{pmatrix}\sim\begin{pmatrix}\boldsymbol{P}_3 & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{P}_4\end{pmatrix}.$$

步骤 5. 对每一个等价类中的代表元,输出 4 轮超级 S 盒下扩散层的分支数为 3 的置换 $\boldsymbol{P}$.

此时,当 $\boldsymbol{PL}$,$\boldsymbol{PR}$ 其中一个为恒等置换时,不存在 MDS 矩阵.所以我们进一步将 $\boldsymbol{PL}$,$\boldsymbol{PR}$ 其中一个放宽为循环移位,此时,满足 2 轮全扩散且超级扩散层为 MDS 时的等价类共有 3 556 个.部分具体实例在附录中给出.这一结果显示 uBlock-128 算法选择的向量置换是最优的,不存在可进一步减少指令数的最优向量置换对.受益于我们的等价类划分方法,最优向量置换对的数量大幅减少,这对后续进一步筛选满足其他安全性指标时提供了便利.

3.2 256 b 分组

256 b 分组的 uBlock 类结构的扩散层描述如图 6 所示.其中,

$$\boldsymbol{T}_1=(1,5,2,6,3,7,4,8),$$
$$\boldsymbol{T}_2=(1,3,5,7,2,4,6,8),$$
$$\boldsymbol{M}_{256}=\begin{pmatrix}\boldsymbol{PL} & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{PR}\end{pmatrix}\begin{pmatrix}\boldsymbol{A}_4 & \boldsymbol{B}_4\\ \boldsymbol{B}_4 & \boldsymbol{C}_4\end{pmatrix}=\begin{pmatrix}\boldsymbol{PLA}_4 & \boldsymbol{PLB}_4\\ \boldsymbol{PRB}_4 & \boldsymbol{PRC}_4\end{pmatrix}.$$

我们首先关注 256 b 分组的 uBlock 类结构的全扩散轮数,得到定理 1.

定理 1. 对于 256 b 分组的 uBlock 类结构,$\boldsymbol{PL}$,$\boldsymbol{PR}$ 选择基于字节的向量置换时,其全扩散轮数的下界为 3.

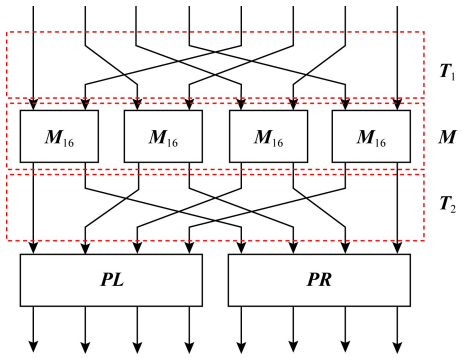


Fig. 6 Diffusion layer of uBlock-like structure with 256 bit sizes

图 6 256 b 分组的 uBlock 类结构的扩散层

证明. 将 256 b 分组的 uBlock 类结构看作是 8 个分支的输入,从左到右依次是第 1 到第 8 个分支,每个分支都为 8 b.我们假定输入向量汉明重量为 1,为(0,1,0,…,0),则经过一轮轮函数的输出为:

第 1 个分支为$(\boldsymbol{a}_1,\boldsymbol{a}_2,\boldsymbol{a}_3,\boldsymbol{a}_4)$,其中

$$\boldsymbol{a}_1=(0,0),\boldsymbol{a}_3=(0,1),\boldsymbol{a}_2=\boldsymbol{a}_4=(1,1).$$

第 5 个分支为$(\boldsymbol{b}_1,\boldsymbol{b}_2,\boldsymbol{b}_3,\boldsymbol{b}_4)$,其中

$$\boldsymbol{b}_1=\boldsymbol{b}_2=(1,1),\boldsymbol{b}_3=\boldsymbol{b}_4=(1,0).$$

其余分支的汉明重量均为零.注意到,$(\boldsymbol{a}_1,\boldsymbol{a}_2,\boldsymbol{a}_3,\boldsymbol{a}_4)$中的元素只能置换到下一轮 $\boldsymbol{M}_{16}$ 输入的左半支,$(\boldsymbol{b}_1,\boldsymbol{b}_2,\boldsymbol{b}_3,\boldsymbol{b}_4)$中的元素只能置换到右半支.然而,要使 2 轮全扩散,进入下一轮 4 个 $\boldsymbol{M}_{16}$ 的向量汉明重量只能为(2,2,3,4)和(2,3,3,3),且汉明重量为 2 的部分只能有 4 种搭配,即$(\boldsymbol{a}_1,\boldsymbol{b}_1)$,$(\boldsymbol{a}_1,\boldsymbol{b}_2)$,$(\boldsymbol{a}_3,\boldsymbol{b}_3)$和$(\boldsymbol{a}_3,\boldsymbol{b}_4)$.然而由第 2 节可知,对于 $\boldsymbol{M}_{16}$,输入向量汉明重量为 2 时,只有 8 种情形可以全扩散,这 8 种情形左右半支汉明重量均为 1,且非零位置只有(1,0)与(1,0)搭配和(0,1)与(0,1)搭配.因此,256 b 分组的 uBlock 类结构至少需要 3 轮全扩散.

由表 3 可知,若 256 b 分组的 uBlock 类结构的 $\boldsymbol{P}$ 层使用更加细粒度的置换,则可能会在 2 轮达到全扩散.考虑到在算法实现时大置换实现效率不佳,因此我们并未尝试寻找大置换下 2 轮达到全扩散的情形.

接下来,我们试图寻找 3 轮全扩散下 256 b 分组的 uBlock 类结构的最优向量置换.对于 $\boldsymbol{PL}$ 和 $\boldsymbol{PR}$,我们与 uBlock-256 一样,考虑面向字节的向量置换.考虑到软件性能的提升,我们假定 $\boldsymbol{PL}$ 或 $\boldsymbol{PR}$ 其中一个为恒等变换,此时我们找到大量满足 3 轮全扩散的置换,在附录中我们给出了部分实例.

此外,我们考虑 4 轮超级 S 盒下,扩散层的分支数为 4 的情形.由于

$$\boldsymbol{M}_{\text{super}}=\boldsymbol{T}_2\cdot\begin{pmatrix}\boldsymbol{PLA}_2\boldsymbol{PL} & \boldsymbol{PLB}_2\boldsymbol{PR}\\ \boldsymbol{PRB}_2\boldsymbol{PL} & \boldsymbol{PRC}_2\boldsymbol{PR}\end{pmatrix}\cdot\boldsymbol{T}_1,$$

我们可将扩散层看作一个 4×4 的分块矩阵,考虑其分支数为 4 的情形.此时,根据超级 S 盒和分支数的概念,可知其 4 轮至少有 32 个活跃 S 盒(与 uBlock-256 算法中的置换在 4 轮时的活跃 S 盒一致).

我们考虑 $\boldsymbol{Q}$ 的形式为 $\begin{pmatrix}\boldsymbol{Q}_1 & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{Q}_2 & \boldsymbol{O} & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{Q}_3 & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{Q}_4\end{pmatrix}$.若要使 $\boldsymbol{QM}=\boldsymbol{MQ}$,则 $\boldsymbol{Q}$ 需满足条件:

$$\boldsymbol{Q}_1\boldsymbol{A}_2=\boldsymbol{A}_2\boldsymbol{Q}_1,\quad \boldsymbol{Q}_2\boldsymbol{A}_2=\boldsymbol{A}_2\boldsymbol{Q}_2,$$
$$\boldsymbol{Q}_3\boldsymbol{C}_2=\boldsymbol{C}_2\boldsymbol{Q}_3,\quad \boldsymbol{Q}_4\boldsymbol{C}_2=\boldsymbol{C}_2\boldsymbol{Q}_4,$$
$$\boldsymbol{Q}_1\boldsymbol{B}_2=\boldsymbol{B}_2\boldsymbol{Q}_3,\quad \boldsymbol{Q}_2\boldsymbol{B}_2=\boldsymbol{B}_2\boldsymbol{Q}_4,$$
$$\boldsymbol{Q}_3\boldsymbol{B}_2=\boldsymbol{B}_2\boldsymbol{Q}_1,\quad \boldsymbol{Q}_4\boldsymbol{B}_2=\boldsymbol{B}_2\boldsymbol{Q}_2.$$

因此,我们给出如下搜索策略 3:

策略 3. uBlock-256 最优向量置换的搜索策略.

步骤 1. 寻找使得 $\boldsymbol{Q}_3\cdot\boldsymbol{A}_2=\boldsymbol{A}_2\cdot\boldsymbol{Q}_3$ 的所有置换 $\boldsymbol{Q}_3$,记为集合 $S_{\boldsymbol{Q}_3}$;寻找使得 $\boldsymbol{Q}_4\cdot\boldsymbol{C}_2=\boldsymbol{C}_2\cdot\boldsymbol{Q}_4$ 的所有置换 $\boldsymbol{Q}_4$,记为集合 $S_{\boldsymbol{Q}_4}$.

步骤 2. 寻找集合对 $S_{\boldsymbol{Q}_a,\boldsymbol{Q}_b}$,使得 $\boldsymbol{Q}_a\in S_{\boldsymbol{Q}_3}$,$\boldsymbol{Q}_b\in S_{\boldsymbol{Q}_4}$ 满足 $\boldsymbol{Q}_a\cdot\boldsymbol{B}_2=\boldsymbol{B}_2\cdot\boldsymbol{Q}_b$;寻找集合对 $S_{\boldsymbol{Q}_c,\boldsymbol{Q}_d}$,使得 $\boldsymbol{Q}_c\in S_{\boldsymbol{Q}_3}$,$\boldsymbol{Q}_d\in S_{\boldsymbol{Q}_4}$ 满足 $\boldsymbol{Q}_d\cdot\boldsymbol{B}_2=\boldsymbol{B}_2\cdot\boldsymbol{Q}_c$.

步骤 3. 取 $(\boldsymbol{Q}_1,\boldsymbol{Q}_2)\in S_{\boldsymbol{Q}_a,\boldsymbol{Q}_b}$,$(\boldsymbol{Q}_3,\boldsymbol{Q}_4)\in S_{\boldsymbol{Q}_c,\boldsymbol{Q}_d}$,对 $\boldsymbol{P}$ 的全空间做等价类划分,即若

$$\begin{pmatrix}\boldsymbol{Q}_1 & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{Q}_2 & \boldsymbol{O} & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{Q}_3 & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{Q}_4\end{pmatrix}\begin{pmatrix}\boldsymbol{P}_1 & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{P}_2\end{pmatrix}\cdot$$
$$\begin{pmatrix}\boldsymbol{Q}_1^{-1} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{Q}_2^{-1} & \boldsymbol{O} & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{Q}_3^{-1} & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{Q}_4^{-1}\end{pmatrix}=\begin{pmatrix}\boldsymbol{P}_3 & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{P}_4\end{pmatrix},$$

则 $\begin{pmatrix}\boldsymbol{P}_1 & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{P}_2\end{pmatrix}\sim\begin{pmatrix}\boldsymbol{P}_3 & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{P}_4\end{pmatrix}$.

步骤 4. 对每一个等价类中的代表元,输出 3 轮全扩散且 4 轮超级 S 盒下扩散层的分支数为 4 的置换 $\boldsymbol{P}$.

经过实验,当 $\boldsymbol{PL}$ 和 $\boldsymbol{PR}$ 其中一个为恒等变换时,我们并未找到 3 轮全扩散且在 4 轮超级扩散层分支数为 4 的置换对.对于 $\boldsymbol{PL}$ 和 $\boldsymbol{PR}$ 均为一般置换时,存在大量 3 轮全扩散且在 4 轮超级扩散层分支数为 4 的置换对,部分具体实例在附录中给出.

在仅考虑全扩散轮数、性能和超级扩散层的分支数这 3 个指标时,我们的结果与 uBlock-256 使用的向量置换是一致的.然而,本文中给出的等价类划分规则可以将满足条件的置换对缩小到较小的范围,这将为后续进一步测评其抵抗其他分析方法时提供便利.

4 总 结

在本文中,我们探索 uBlock 类结构最优向量置换的选取.对于 uBlock 算法族,扩散层分为 2 部分:一部分是由 Feistel 结构构造的 16 维最优二元扩散层,另一部分为 2 个向量置换的并置.我们的目标是在二元扩散层固定的前提下,寻找最优向量置换来提升算法整体的安全性和实现效率.本文中,我们主要的评价指标为算法的全扩散轮数、软件性能和 4 轮超级 S 盒下扩散层的分支数.

首先,我们探索 uBlock 类结构的扩散性,给出了不同规模下 uBlock 类结构全扩散轮数的下界.其次,为了减少搜索复杂度,基于 uBlock 类结构的特点,我们提出了等价类划分技术.进一步,基于全扩散轮数最优、软件性能优良和超级扩散层的分支数,我们设计了 uBlock 类结构最优向量置换的搜索策略.最后,将搜索策略应用于 128 b 和 256 b 分组的 uBlock 类结构中.在具体应用时,结合不同分组长度下算法的特点,我们进一步优化了最优向量置换的搜索策略,并给出具体实例.

对于 128 b 分组的 uBlock 类结构,若 $\boldsymbol{PL}$ 和 $\boldsymbol{PR}$ 其中一个为恒等变换时,存在满足 2 轮全扩散的置换对;然而,不存在 2 轮全扩散且超级扩散层分支数最优的置换对.因此,我们将 $\boldsymbol{PL}$ 或 $\boldsymbol{PR}$ 放宽到循环移位后,给出了全扩散轮数及超级扩散层分支数均最优的置换对.对于 256 b 分组的 uBlock 类结构,我们证明了其最优全扩散轮数为 3 轮,若 $\boldsymbol{PL}$ 或 $\boldsymbol{PR}$ 其中一个为恒等变换时,存在 3 轮全扩散的置换对,然而不存在全扩散轮数为 3 且超级扩散层的分支数为 4 的置换对.与 uBlock 算法相比,在仅考虑全扩散轮数这一指标时,算法所需的向量置换指令更少,从而有更高的软件实现效率.在考虑全扩散轮数和超级扩散层分支数 2 个指标后,我们的结果与 uBlock 算法中使用的向量置换是一致的,表明 uBlock 算法选择的向量置换是最优的,不存在可进一步减少指令

数的向量置换对,但是我们提出的等价类划分规则可以将置换对的数量大幅减少,为后续 uBlock 类算法的设计提供技术支持.

作者贡献申明:李晓丹提出了算法思路,撰写论文;吴文玲提出指导意见并修改论文;张丽提出修改意见.

参 考 文 献

[1] Daemen J, Rijmen V. The Design of Rijndael [M]. 2nd eds. Berlin: Springer, 2002

[2] Kwon D, Kim J, Park S, et al. New block cipher: ARIA [C] //Proc of the 6th Int Conf on Information Security and Cryptology(ICISC 2003). Berlin: Springer, 2003: 432-445

[3] Biryukov A. Analysis of involutional ciphers: Khazad and Anubis [C] //Proc of Int Workshop on Fast Software Encryption. Berlin: Springer, 2003: 45-53

[4] Guo Jian, Peyrin T, Poschmann A, et al. The LED block cipher [C] //Proc of Int Workshop on Cryptographic Hardware and Embedded Systems. Berlin: Springer, 2011: 326-341

[5] Banik S, Bogdanov A, Isobe T, et al. Midori: A block cipher for low energy [C] //Proc of the 21st Int Conf on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2015). Berlin: Springer, 2015: 411-436

[6] Guo Jian, Peyrin T, Poschmann A. The PHOTON family of lightweight Hash functions [C] //Proc of the 31st Annual Int Cryptology Conf (CRYPTO 2011). Berlin: Springer, 2011: 222-239

[7] Avanzi R. The QARMA block cipher family [J]. IACR Transactions on Symmetric Cryptology, 2017, 2017(1): 4-44

[8] Beierle C, Jean J, Kölbl S, et al. The SKINNY family of block ciphers and its low-latency variant MANTIS [C] //Proc of the 36th Annual Int Cryptology Conf (CRYPTO 2016). Berlin: Springer, 2016: 123-153

[9] Kitsos P, Koufopavlou O. Efficient architecture and hardware implementation of the Whirlpool Hash function [J]. IEEE Transactions on Consumer Electronics, 2004, 50(1): 208-213

[10] Daemen J. Cipher and Hash function design strategies based on linear and differential cryptanalysis [D]. Belgium: KU Leuven, 1995

[11] Beierle C, Jovanovic P, Lauridsen M M, et al. Analyzing permutations for AES-like ciphers: Understanding ShiftRows [C] //Proc of Topics in Cryptology, the Cryptographer's Track at the RSA Conf 2015. Berlin: Springer, 2015: 37-58

[12] Todo Y, Aoki K. Wide trail design strategy for binary MixColumns-enhancing lower bound of number of active S-boxes [C] //Proc of the 14th Int Conf on Applied Cryptography and Network Security (ACNS 16). Berlin: Springer, 2016: 467-484

[13] Alfarano G N, Beierle C, Isobe T, et al. ShiftRows alternatives for AES-like ciphers and optimal cell permutations for Midori and SKINNY [J]. IACR Transactions on Symmetric Cryptology, 2018, 2018(2): 20-47

[14] Wu Wenling, Zhang Lei, Zheng Yafei, et al. The block cipher uBlock [J]. Journal of Cryptologic Research, 2019, 6(6): 690-703 (in Chinese)
(吴文玲, 张蕾, 郑雅菲, 等. 分组密码 uBlock [J]. 密码学报, 2019, 6(6): 690-703)

[15] Shannon C E. Communication theory of secrecy systems [J]. Bell Systems Technical Journal, 1949, 28(4): 656-715

[16] Suzaki T, Minematsu K. Improving the generalized Feistel [C] //Proc of the 17th Int Workshop on Fast Software Encryption (FSE 2010). Berlin: Springer, 2010: 19-39

Li Xiaodan, born in 1992. PhD. Her main research interests include design and cryptanalysis of block ciphers.

李晓丹,1992 年生.博士.主要研究方向为分组密码的设计和密码分析.

Wu Wenling, born in 1966. PhD, professor, and PhD supervisor. Senior member of CCF. Her main research interests include design and cryptanalysis of block ciphers and Hash functions, and cryptography.

吴文玲,1966 年生.博士,教授,博士生导师.CCF 高级会员.主要研究方向为分组密码和哈希函数的设计和密码分析以及密码学.

Zhang Li, born in 1994. PhD candidate. Her main research interests include design and cryptanalysis of block ciphers.

张 丽,1994 年生.博士研究生.主要研究方向为分组密码的设计和密码分析.

附录:

在附表 1 中,我们给出了 128 b 及 256 b 分组的 uBlock 类结构的部分最优向量置换.其中,128 b 分组的 uBlock 类结构的最优向量置换指的是 2 轮达到全扩散,且在 4 轮超级 S 盒下是 2 维 MDS 矩阵;256 b 分组的 uBlock 类结构的最优向量置换指的是 3 轮达到全扩散,且在 4 轮超级 S 盒下是 4 维 Near-MDS 矩阵. 为了方便,附表 1 和附表 2 中我们用一个大置换来表示最优置换,并未将 ***PL*** 和 ***PR*** 分别罗列出来.

Table 1 Some Optimal Permutations for uBlock-like Structure with 128 bit Block Sizes

附表 1 128 b 分组的 uBlock 类结构的部分最优置换

PL 取循环移位时	***PR*** 取循环移位时
(3, 4, 5, 6, 7, 0, 1, 2, 8, 9, 10, 12, 13, 14, 11, 15)	(0, 1, 2, 6, 3, 4, 5, 7, 13, 14, 15, 8, 9, 10, 11, 12)
(3, 4, 5, 6, 7, 0, 1, 2, 8, 9, 10, 12, 14, 13, 11, 15)	(0, 1, 2, 7, 3, 4, 5, 6, 13, 14, 15, 8, 9, 10, 11, 12)
(3, 4, 5, 6, 7, 0, 1, 2, 8, 9, 10, 12, 14, 15, 11, 13)	(0, 1, 2, 7, 3, 4, 6, 5, 13, 14, 15, 8, 9, 10, 11, 12)
(3, 4, 5, 6, 7, 0, 1, 2, 8, 9, 10, 12, 15, 14, 11, 13)	(0, 1, 2, 6, 3, 5, 4, 7, 13, 14, 15, 8, 9, 10, 11, 12)
(3, 4, 5, 6, 7, 0, 1, 2, 8, 9, 10, 15, 12, 13, 11, 14)	(0, 1, 2, 7, 3, 5, 4, 6, 13, 14, 15, 8, 9, 10, 11, 12)
(3, 4, 5, 6, 7, 0, 1, 2, 8, 9, 10, 14, 13, 12, 11, 15)	(0, 1, 2, 6, 3, 7, 4, 5, 13, 14, 15, 8, 9, 10, 11, 12)
(3, 4, 5, 6, 7, 0, 1, 2, 8, 9, 10, 15, 13, 12, 11, 14)	(0, 1, 2, 6, 3, 7, 5, 4, 13, 14, 15, 8, 9, 10, 11, 12)
(3, 4, 5, 6, 7, 0, 1, 2, 8, 9, 10, 14, 15, 12, 11, 13)	(0, 1, 2, 7, 3, 6, 5, 4, 13, 14, 15, 8, 9, 10, 11, 12)
(3, 4, 5, 6, 7, 0, 1, 2, 8, 9, 10, 15, 14, 12, 11, 13)	(0, 1, 2, 7, 4, 3, 5, 6, 13, 14, 15, 8, 9, 10, 11, 12)
(3, 4, 5, 6, 7, 0, 1, 2, 8, 9, 10, 15, 13, 14, 11, 12)	(0, 1, 2, 6, 4, 3, 7, 5, 13, 14, 15, 8, 9, 10, 11, 12)
(3, 4, 5, 6, 7, 0, 1, 2, 8, 9, 10, 15, 14, 13, 11, 12)	(0, 1, 2, 7, 4, 3, 6, 5, 13, 14, 15, 8, 9, 10, 11, 12)
(3, 4, 5, 6, 7, 0, 1, 2, 8, 9, 10, 12, 13, 14, 15, 11)	(0, 1, 2, 7, 6, 3, 4, 5, 13, 14, 15, 8, 9, 10, 11, 12)
(3, 4, 5, 6, 7, 0, 1, 2, 8, 9, 10, 12, 13, 15, 14, 11)	(0, 1, 2, 7, 6, 3, 5, 4, 13, 14, 15, 8, 9, 10, 11, 12)
(3, 4, 5, 6, 7, 0, 1, 2, 8, 9, 10, 12, 14, 13, 15, 11)	(0, 1, 2, 7, 4, 5, 3, 6, 13, 14, 15, 8, 9, 10, 11, 12)
(3, 4, 5, 6, 7, 0, 1, 2, 8, 9, 10, 12, 15, 14, 13, 11)	(0, 1, 2, 6, 5, 4, 3, 7, 13, 14, 15, 8, 9, 10, 11, 12)
(3, 4, 5, 6, 7, 0, 1, 2, 8, 9, 10, 13, 12, 14, 15, 11)	(0, 1, 2, 7, 5, 4, 3, 6, 13, 14, 15, 8, 9, 10, 11, 12)
(3, 4, 5, 6, 7, 0, 1, 2, 8, 9, 10, 13, 12, 15, 14, 11)	(0, 1, 2, 7, 6, 4, 3, 5, 13, 14, 15, 8, 9, 10, 11, 12)
(3, 4, 5, 6, 7, 0, 1, 2, 8, 9, 10, 14, 12, 13, 15, 11)	(0, 1, 2, 6, 5, 7, 3, 4, 13, 14, 15, 8, 9, 10, 11, 12)
(3, 4, 5, 6, 7, 0, 1, 2, 8, 9, 10, 15, 12, 13, 14, 11)	(0, 1, 2, 7, 5, 6, 3, 4, 13, 14, 15, 8, 9, 10, 11, 12)
(3, 4, 5, 6, 7, 0, 1, 2, 8, 9, 10, 15, 12, 14, 13, 11)	(0, 1, 2, 6, 4, 5, 7, 3, 13, 14, 15, 8, 9, 10, 11, 12)
(3, 4, 5, 6, 7, 0, 1, 2, 8, 9, 10, 14, 13, 12, 15, 11)	(0, 1, 2, 7, 4, 5, 6, 3, 13, 14, 15, 8, 9, 10, 11, 12)
(3, 4, 5, 6, 7, 0, 1, 2, 8, 9, 10, 14, 15, 12, 13, 11)	(0, 1, 2, 7, 4, 6, 5, 3, 13, 14, 15, 8, 9, 10, 11, 12)
(3, 4, 5, 6, 7, 0, 1, 2, 8, 9, 10, 13, 14, 15, 12, 11)	(0, 1, 2, 6, 5, 4, 7, 3, 13, 14, 15, 8, 9, 10, 11, 12)
(3, 4, 5, 6, 7, 0, 1, 2, 8, 9, 10, 13, 15, 14, 12, 11)	(0, 1, 2, 6, 7, 4, 5, 3, 13, 14, 15, 8, 9, 10, 11, 12)
(3, 4, 5, 6, 7, 0, 1, 2, 8, 9, 10, 14, 13, 15, 12, 11)	(0, 1, 2, 7, 6, 4, 5, 3, 13, 14, 15, 8, 9, 10, 11, 12)
(3, 4, 5, 6, 7, 0, 1, 2, 8, 9, 10, 15, 13, 14, 12, 11)	(0, 1, 2, 6, 5, 7, 4, 3, 13, 14, 15, 8, 9, 10, 11, 12)
(3, 4, 5, 6, 7, 0, 1, 2, 8, 9, 10, 15, 14, 13, 12, 11)	(0, 1, 2, 6, 7, 5, 4, 3, 13, 14, 15, 8, 9, 10, 11, 12)
(3, 4, 5, 6, 7, 0, 1, 2, 8, 9, 12, 10, 13, 14, 11, 15)	(0, 1, 2, 7, 6, 5, 4, 3, 13, 14, 15, 8, 9, 10, 11, 12)
(3, 4, 5, 6, 7, 0, 1, 2, 8, 9, 12, 10, 14, 13, 11, 15)	(0, 1, 7, 2, 3, 4, 5, 6, 13, 14, 15, 8, 9, 10, 11, 12)
(3, 4, 5, 6, 7, 0, 1, 2, 8, 9, 12, 10, 14, 15, 11, 13)	(0, 1, 7, 2, 3, 4, 6, 5, 13, 14, 15, 8, 9, 10, 11, 12)

Table 2 Some Optimal Permutations for uBlock-like Structure with 256 bit Block Sizes

附表 2 256 b 分组的 uBlock 类结构的部分最优置换

256 b 分组的 uBlock 类结构的部分最优置换
(0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 16, 20, 24, 28, 21, 17, 25, 29, 18, 22, 26, 30, 19, 23, 27, 31)
(0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 16, 20, 24, 28, 21, 17, 25, 29, 18, 22, 26, 30, 31, 23, 27, 19)
(0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 16, 20, 24, 28, 21, 17, 25, 29, 18, 22, 30, 26, 19, 23, 27, 31)
(0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 16, 20, 24, 28, 21, 17, 25, 29, 18, 22, 30, 26, 31, 23, 27, 19)
(0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 16, 20, 24, 28, 21, 17, 25, 29, 18, 26, 22, 30, 19, 23, 27, 31)
(0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 16, 20, 24, 28, 21, 17, 25, 29, 18, 26, 22, 30, 31, 23, 27, 19)
(0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 16, 20, 24, 28, 21, 17, 25, 29, 22, 18, 26, 30, 19, 23, 27, 31)
(0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 16, 20, 24, 28, 21, 17, 25, 29, 22, 18, 26, 30, 23, 19, 27, 31)
(0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 16, 20, 24, 28, 21, 17, 25, 29, 22, 18, 26, 30, 31, 23, 27, 19)
(0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 16, 20, 24, 28, 21, 17, 25, 29, 22, 26, 30, 18, 19, 23, 27, 31)
(0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 16, 20, 24, 28, 21, 17, 25, 29, 22, 26, 30, 18, 23, 19, 27, 31)
(0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 16, 20, 24, 28, 21, 17, 25, 29, 22, 26, 30, 18, 31, 23, 27, 19)
(0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 16, 20, 24, 28, 25, 17, 21, 29, 18, 22, 26, 30, 19, 23, 27, 31)
(0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 16, 20, 24, 28, 25, 17, 21, 29, 18, 22, 26, 30, 23, 19, 27, 31)
(0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 16, 20, 24, 28, 25, 17, 21, 29, 18, 22, 26, 30, 31, 23, 27, 19)
(0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 16, 20, 24, 28, 25, 17, 21, 29, 18, 22, 30, 26, 19, 23, 27, 31)
(0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 16, 20, 24, 28, 25, 17, 21, 29, 18, 22, 30, 26, 23, 19, 27, 31)
(0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 16, 20, 24, 28, 25, 17, 21, 29, 18, 22, 30, 26, 31, 23, 27, 19)
(0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 16, 20, 24, 28, 25, 17, 21, 29, 18, 26, 22, 30, 19, 23, 27, 31)
(0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 16, 20, 24, 28, 25, 17, 21, 29, 18, 26, 22, 30, 23, 19, 27, 31)
(0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 16, 20, 24, 28, 25, 17, 21, 29, 18, 26, 22, 30, 31, 23, 27, 19)
(0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 16, 20, 24, 28, 25, 17, 21, 29, 22, 18, 26, 30, 19, 23, 27, 31)
(0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 16, 20, 24, 28, 25, 17, 21, 29, 22, 18, 26, 30, 23, 19, 27, 31)
(0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 16, 20, 24, 28, 25, 17, 21, 29, 22, 18, 26, 30, 31, 23, 27, 19)
(0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 16, 20, 24, 28, 29, 17, 21, 25, 18, 22, 26, 30, 19, 23, 27, 31)
(0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 16, 20, 24, 28, 29, 17, 21, 25, 18, 22, 26, 30, 23, 19, 27, 31)
(0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 16, 20, 24, 28, 29, 17, 21, 25, 18, 22, 26, 30, 31, 23, 27, 19)
(0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 16, 20, 24, 28, 29, 17, 21, 25, 18, 22, 30, 26, 19, 23, 27, 31)
(0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 16, 20, 24, 28, 29, 17, 21, 25, 18, 22, 30, 26, 23, 19, 27, 31)
(0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 16, 20, 24, 28, 29, 17, 21, 25, 18, 22, 30, 26, 31, 23, 27, 19)