

使用差分概率分布表对差分分析的研究

刘 慧^{1,2}, 杨 理^{1,2}

¹ 中国科学院信息工程研究所信息安全国家重点实验室 北京 中国 100093

² 中国科学院大学网络空间安全学院 北京 中国 100049

摘要 差分分析是分析分组密码安全性的最有效的方法之一,本文从一个新的角度研究了轮函数为双射的密钥交替的迭代型分组密码的差分性质。S 盒的差分分布表在研究迭代型分组密码的差分分析中起到了重要作用。类比 S 盒的差分分布表,我们研究轮函数的差分概率分布表(DPT),我们发现当迭代型分组密码的轮函数是双射时,其 DPT 矩阵是一个马尔可夫矩阵,并通过研究 DPT 矩阵的性质来研究分组密码的差分性质。一方面,我们通过分析矩阵的性质给出了一些有关迭代型分组密码的差分性质的证明,并且研究了当轮函数是对合轮函数时其最大差分概率的性质。我们证明了对对合轮函数的迭代型分组密码来说,一定存在一个形如 $\alpha \rightarrow \alpha$ 的最大差分概率。我们还证明了若迭代型分组密码的 DPT 矩阵有两个值为 1 的特征值,那么经过足够高的轮数后,该分组密码一定没有高概率差分;而当值为 1 的特征值个数多于两个时,其可能存在任意轮的高概率差分。另一方面,由于对实际的迭代型分组密码来说,轮函数的差分概率分布表所对应的矩阵维数过高,实现这样规模的矩阵的存储与计算并不可行。为了解决该问题,我们分析其字节级截断差分概率,构造其字节级截断差分概率分布表,通过研究其对应的矩阵来研究字节级截断差分性质。对于状态被分成 16 个字节或半字节的 r 轮分组密码来说,构造该表的空间复杂度是 $c_1 \cdot 2^{32}$,时间复杂度是 $c_2 \cdot 2^{37.92}$,其中 $c_1, c_2 \in O(\log_2 r)$ 。现阶段,可以借助高性能计算机在可接受的时间内得到整个分组密码的全部字节级截断差分信息。

关键词 分组密码; 差分分析; 截断差分分析; 差分分布表; 对合轮函数
中图法分类号 TN918.1 **DOI 号** 10.19363/J.cnki.cn10-1380/tn.2025.09.13

A New Sight of Differential Cryptanalysis using DDT

LIU Hui^{1,2}, YANG Li^{1,2}

¹ State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China

² School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

Abstract Differential cryptanalysis is one of the most effective methods of block cipher cryptanalysis. In this paper we study the differential properties of key-alternating iterated block cipher with bijective round function from a new sight. Differential Distribution Table(DDT) plays an important role in the differential cryptanalysis of iterated block cipher. By analogy with DDT, we study the Differential Probability Distribution Table(DPT) of round function, and we find the DPT matrix is a Markov matrix when the iterated block cipher is bijective. Then we study the differential properties of block cipher by analyzing the properties of DPT matrix. On one hand, we provide some proofs concerning the differential properties of block cipher and we focus on the special case when the round function is involutory. We proved that there must exist a maximum differential probability in the form of $\alpha \rightarrow \alpha$ for block ciphers with involutory round function. We also proved that if the DPT matrix of a block cipher's round function has only two eigenvalues with value 1, then the block cipher must have no high-probability differential after sufficiently many rounds. However, when the number of DPT matrix's eigenvalue with value 1 exceeds 2, the block cipher may have a high probability difference for any round. On the other hand, for a practical block cipher, the dimension of its round function's DPT matrix is so high that storing and computing such matrix is infeasible. To settle this problem, we analyze the byte-oriented truncated differential probability and construct Byte-oriented Truncated Differential Probability Distribution Table(TPT) of iterated block cipher to study the byte-oriented truncated differential properties. For a r -round block cipher the block of which can be divided into 16 bytes or nibbles, it costs $c_1 \cdot 2^{32}$ in memory and takes $c_2 \cdot 2^{37.92}$ in time to construct TPT, where $c_1, c_2 \in O(\log_2 r)$. We can get all the byte-oriented truncated differential information of the whole block cipher with high-performance computer in acceptable time at the present stage.

Key words block cipher; differential cryptanalysis; truncated differential cryptanalysis; differential distribution table; involutory round function

通讯作者: 刘慧, 硕士生, Email: liuhui0524@iie.ac.cn。

本课题得到国家自然科学基金项目(No. 61672517)资助。

收稿日期: 2020-10-31; 修改日期: 2021-02-20; 定稿日期: 2023-02-24

1 引言

差分分析^[1]由 Biham 和 Shamir 于 1991 年提出, 此后便成为分析分组密码安全性最有效的方法之一。其基本思想是: 首先选择一定数量的输入差分为某一定值的明文对, 获取他们相应的密文对; 然后计算这些密文对的差分, 如果某个密文差分出现的概率较高则我们就可以将这个密码算法与随机置换区分开来, 进而可以对密码算法进行攻击。在差分分析中, 最关键的步骤就是要找到高概率差分。

在大多数迭代型分组密码中, S 盒是其轮函数中唯一的非线性元件。因此在差分分析中我们主要研究的就是 S 盒的差分性质。差分分布表(Differential Distribution Table, DDT)是目前研究 S 盒差分性质最有力的方法之一, 其能够完全地包含一个 S 盒的全部差分性质。对于一个 n 比特的 S 盒, 其 DDT 可以看成是一个 2^n 阶方阵(下面将该矩阵称为 DDT 矩阵)。为了方便直接地得到差分概率, 我们将每个矩阵的元素除以 2^n , 得到差分概率分布表(Differential Probability Distribution Table, DPT), 该表的第 i 行第 j 列元素表示输入差分为 i , 输出差分为 j 的差分概率。

在亚密会 2018 中, Tim Beyne 提出了一种使用线性相关矩阵来研究分组密码的不变子空间攻击的方法^[2]。另外 Kim 等人^[3]通过构造加密特征矩阵, 并使用矩阵和向量的乘法对不可能差分进行了分析。受到他们的启发, 我们考虑使用矩阵的方法, 即分组密码的 DPT 矩阵来研究其差分性质。SPN 结构的迭代型分组密码, 如 AES^[4]、CRAFT^[5]、MIDORI^[6]等, 每轮使用相同或相似的公开轮函数。类比 S 盒的 DPT 的建立, 如果我们能建立整个轮函数的 DPT, 就能够得到轮函数的差分概率分布, 且在满足马尔可夫密码和随机等价假设的前提下, 这样得到的差分概率是准确的。本文主要利用 DPT 矩阵和马尔可夫矩阵的性质来研究差分分析。

Lai 等人^[7]提出了马尔可夫密码的概念, 认为在每轮的密钥都是独立且均匀随机分布时, 一个差分序列可以看成是一个马尔可夫链, 那么密码算法的第 $r+1$ 轮的差分状态只与第 r 轮的状态有关, 这样就可以通过研究轮函数的概率转移矩阵来分析 r 轮马尔可夫密码的差分性质。在这个基础上, Hornauer 等人^[8]利用马尔可夫链和对称群的性质证明了当轮数足够高时, 在满足马尔可夫密码和随机等价假设的

前提下, DES 是能够抵抗差分攻击的。在 1994 年的亚密会上, Luke 等人^[9]使用马尔可夫链进一步地对差分分析和线性分析进行了研究, 并说明了当轮数足够高时, 大多数迭代型分组密码都能抵抗差分攻击, 但对于一个给定的轮函数, 很难判断当轮数足够多时整个密码算法的差分性质。这些工作都是将差分序列看成是一个马尔可夫链来进行的。在本文中, 我们提出一个新的角度去分析马尔可夫密码, 我们发现轮函数的概率转移矩阵就是它的 DPT 矩阵, 且 DPT 矩阵以及它的转置都是一个马尔可夫矩阵。我们研究这个马尔可夫矩阵的性质, 进一步得到密码算法的差分性质。

对于实际的分组密码算法, 其分组长度至少为 64 比特, 其 DPT 矩阵的阶数至少为 2^{64} 。显然, 以目前计算机的能力, 我们无法储存和处理如此大的矩阵, 因此我们不能直接利用轮函数的 DPT。所以我们研究分组密码的字节级截断差分^[10], 字节级截断差分指我们只关心状态中的每个 cell 或字节上是否有差分, 而不关心差分的值是多少。目前, 比较有效的搜索字节级截断差分的方法是使用基于 MILP 方法的自动化搜索方法^[11]。相比于搜索比特级的差分, 使用 MILP 方法搜索字节级截断差分时将整个 cell 或字节看成是一个单位, 大大减少了搜索规模, 而且这时也不需要考虑 S 盒对截断差分的影响, 因此搜索截断差分特征的速度是非常快的。但随着轮数的增加, 这种方法的搜索时间也会随轮数有较大的增长。此外, 基于 MILP 方法的搜索方法一般只能得到一个或几个差分, 而不是完整的差分闭包, 这样计算得到的差分概率实际上是真实差分概率的一个下界, 要想得到真实的差分概率, 还需要固定首尾差分然后搜索满足这对差分的全部差分特征并计算差分概率, 这往往是非常烦琐和耗时的。使用本文提出的基于矩阵乘法的方法来研究字节级截断差分可以避免上述问题。

本文从 DPT 矩阵的角度研究了差分分析, 将研究差分概率性质的问题转化成了研究矩阵性质的问题。我们发现多轮密码算法的 DPT 矩阵可以由其单个轮函数的 DPT 矩阵的幂次来表示, 并利用矩阵乘法和马尔可夫矩阵的性质研究了迭代型分组密码算法的差分性质。从矩阵特征值的角度提出了一个分组密码抗差分攻击能力的评估方法, 并提出了一种搜索全部准确的字节级截断差分的方法(包括不可能截断差分)。本文的主要贡献包括:

(1) 本文从马尔可夫矩阵的角度研究了轮函数是双射的迭代型分组密码的差分性质。我们发现当

迭代型分组密码的轮函数是双射时, 其 DPT 矩阵是一个马尔可夫矩阵。如果 DPT 矩阵只有两个值为 1 的特征值, 那么经过足够高的轮数后, 其一定不存在高概率差分; 而如果 DPT 矩阵存在多于两个值为 1 的特征值, 那么这个分组密码可能存在任意轮的高概率差分。这个结论可以作为评价 S 盒或轮函数安全性的一种方法。

(2) 本文研究了轮函数为对合轮函数的分组密码的差分性质, 对合轮函数的 DPT 矩阵是一个对称矩阵, 因此我们可以用对称矩阵的一些性质来对该类型的分组密码的差分性质做进一步的分析。我们证明了对于 $r = 2^k$ 轮的该类型的分组密码, 其最大差分概率对应的差分(或之一)一定是 $\alpha \rightarrow \alpha$ 形式的, 此外我们还给出了对不同轮数之间的最差差分概率的关系的估计。

(3) 本文研究了分组密码的字节级截断差分。对于一般的 SPN 结构的迭代型分组密码, 其一般将明文状态分成 16 个 cell 或字节来处理。我们只考虑每个 cell 或字节上是否有差分, 因此我们就可以构造这类分组密码的字节级截断差分概率分布表 (Byte-oriented Truncated Differential Probability Distribution Table, TPT)。这时 TPT 矩阵是一个 2^{16} 阶的方阵, 存储这个矩阵的空间复杂度为 2^{32} , 做一次矩阵乘法的时间复杂度为 $2^{37.92}$ 。对于 r 轮的密码算法只需要进行 $O(\log_2 r)$ 次矩阵乘法即可得到其 TPT 矩阵 T^r 。而且, 当密码算法的轮数 r 增长时, 矩阵乘法计算所需的复杂度的增长也不会很大。我们可以从 TPT 中得到整个密码算法的全部字节级截断差分的准确的差分概率, 而不仅仅是某些高概率差分的概率下界, 并且还能很容易判断是否存在字节级截断不可能差分, 同时也能找到这个截断不可能差分对应的输入和输出差分(TPT 中值为 0 的索引)。此外, 我们还对比了基于 MILP 方法和基于 TPT 矩阵乘法计算方法来搜索字节级截断差分的复杂度。

文章的其余部分安排如下: 第 2 节给出符号说明、一些规定和 DDT、DPT 的构造; 第 3 节使用矩阵和马尔可夫矩阵对差分性质进行一些分析和证明; 第 4 节介绍使用 TPT 矩阵对字节级截断差分的研究; 第 5 节总结全文。

2 符号说明及差分概率分布表

本节介绍相关的预备知识。首先, 我们给出符号说明及研究对象的描述。然后, 我们从 S 盒的 DDT 出发引出 DPT 的概念, 接着介绍了轮函数的 DPT 及

其性质。如果不做特殊说明, 在本文中, 我们研究的 S 盒和轮函数都是双射的。

2.1 符号说明及一些约定

首先, 我们将文中出现的相关符号及其解释列在表 1 中。

表 1 符号说明
Table 1 Notations

符号	解释
DDT	差分分布表
DPT	差分概率分布表
TPT	字节级截断差分概率分布表
$\tilde{D}$	DDT 对应的矩阵
D	DPT 对应的矩阵
T	TPT 对应的矩阵
R	随机置换的 DPT 矩阵
RT	随机置换的 TPT 矩阵
D^r	r 轮分组密码 DPT 矩阵
T^r	r 轮分组密码 TPT 矩阵
E^r	r 轮迭代分组密码
$D[i, j]$	矩阵 D 的第 i 行第 j 列的矩阵元

然后, 我们介绍本文针对的研究对象。本文研究的是轮函数是双射的密钥交替的迭代型分组密码, 且其满足马尔可夫密码和随机等价假设, 如图 1 所示。一般的密钥交替的迭代型分组密码包括若干个相同或相似的轮函数, 这些轮函数由若干个异或密钥操作连接起来, 我们假设每轮的轮函数都相同并记为 F 。并假设每轮的子密钥 $k_0, k_1, \dots, k_{r+1}$ 独立且是随机均匀选取的, 即

$$E^{(r)}: \{0, 1\}^n \rightarrow \{0, 1\}^n$$

$$x \rightarrow F(F \dots (F(x \oplus k_0) \oplus k_1) \dots) \oplus k_{r+1}$$

以下我们简称其为迭代型分组密码。

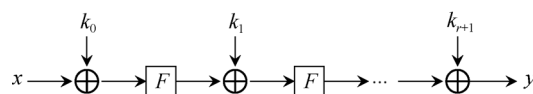


图 1 密钥交替的迭代分组密码

Figure 1 Key-alternating iterated block cipher

2.2 S 盒的 DPT

在大多数迭代型分组密码中, S 盒是分组密码轮函数中唯一的非线性元件。因此为了分析整个分组密码的差分性质, 我们首先需要研究 S 盒的差分性质, S 盒的差分性质由其 DDT 和 DPT 体现。本节我

们引入 S 盒的 DPT 并给出 DPT 的构造。

对于一个 n 比特的 S 盒, 它的 DDT 有 2^n 行和 2^n 列, 其中第 i 行、第 j 列的值表示在穷举了全部 2^n 个数据后, 输入差分为 i 且输出差分为 j 的数据对的个数, 用 A 表示 S 盒的 DDT, 那么差分分布表中元素的值为

$$A[i, j] = \#\{x \in \text{Dom}(S) : S(x) \oplus S(x \oplus i) = j\}$$

其中, $\text{Dom}(S)$ 为 S 盒的定义域。以 CRAFT 的 S 盒为例, 其 S 盒定义如表 2 所示, S 盒的差分分布表如表 3 所示。

表 2 CRAFT 的 S 盒
Table 2 The S-Box of CRAFT

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
S	c	a	d	3	e	b	f	7	8	9	1	5	0	2	4	6

表 3 CRAFT S 盒的 DDT
Table 3 The DDT of CRAFT's S-Box

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	2	4	0	2	2	2	0	2	0	0	0	0	0	2	0
2	0	4	0	0	4	0	0	0	0	4	0	0	4	0	0	0
3	0	0	0	0	2	0	4	2	2	2	0	0	0	2	0	2
4	0	2	4	2	2	2	0	0	2	0	0	2	0	0	0	0
5	0	2	0	0	2	0	0	4	0	2	4	0	2	0	0	0
6	0	2	0	4	0	0	0	2	2	0	0	0	2	2	0	2
7	0	0	0	2	0	4	2	0	0	0	0	2	0	4	2	0
8	0	2	0	2	2	0	2	0	0	2	0	2	2	0	2	0
9	0	0	4	2	0	2	0	0	2	2	0	2	2	0	0	0
a	0	0	0	0	0	4	0	0	0	0	4	0	0	4	0	4
b	0	0	0	0	2	0	0	2	2	2	0	4	0	2	0	2
c	0	0	4	0	0	2	2	0	2	2	0	0	2	0	2	0
d	0	0	0	2	0	0	2	4	0	0	4	2	0	0	2	0
e	0	2	0	0	0	0	0	2	2	0	0	0	2	2	4	2
f	0	0	0	2	0	0	2	0	0	0	4	2	0	0	2	4

从 DDT 我们可以很容易得到某个差分特征的概率。例如, 从表 3 中我们可以知道, 对于 CRAFT 的 S 盒, 输入差分为 $0x2$ 输出差分为 $0xc$ 的概率为 $\frac{4}{16} = 2^{-2}$ 。因此, 对 n 比特 S 盒的 DDT 的每个元素除以 2^n , 就可以得到 DPT 矩阵, 记为 D , D 中元素为

$$D[i, j] = \frac{A[i, j]}{A[0, 0]}$$

D 中的元素 $D[i, j]$ 表示输入差分为 i , 输出差分为 j 的差分特征的概率。CRAFT S 盒的 DPT 如表 4 所示。

表 4 CRAFT S 盒的 DPT

Table 4 The DPT of CRAFT's S-Box

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	$\frac{1}{8}$	$\frac{1}{4}$	0	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	0	$\frac{1}{8}$	0	0	0	0	0	$\frac{1}{8}$	0
2	0	$\frac{1}{4}$	0	0	$\frac{1}{4}$	0	0	0	0	$\frac{1}{4}$	0	0	$\frac{1}{4}$	0	0	0
3	0	0	0	0	$\frac{1}{8}$	0	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	0	0	0	$\frac{1}{8}$	0	$\frac{1}{8}$
4	0	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	0	0	$\frac{1}{8}$	0	0	$\frac{1}{8}$	0	0	0	0
5	0	$\frac{1}{8}$	0	0	$\frac{1}{8}$	0	0	$\frac{1}{4}$	0	$\frac{1}{8}$	$\frac{1}{4}$	0	$\frac{1}{8}$	0	0	0
6	0	$\frac{1}{8}$	0	$\frac{1}{4}$	0	0	0	$\frac{1}{8}$	$\frac{1}{8}$	0	0	0	$\frac{1}{8}$	$\frac{1}{8}$	0	$\frac{1}{8}$
7	0	0	0	$\frac{1}{8}$	0	$\frac{1}{4}$	$\frac{1}{8}$	0	0	0	0	$\frac{1}{8}$	0	$\frac{1}{4}$	$\frac{1}{8}$	0
8	0	$\frac{1}{8}$	0	$\frac{1}{8}$	$\frac{1}{8}$	0	$\frac{1}{8}$	0	0	$\frac{1}{8}$	0	$\frac{1}{8}$	$\frac{1}{8}$	0	$\frac{1}{8}$	0
9	0	0	$\frac{1}{4}$	$\frac{1}{8}$	0	$\frac{1}{8}$	0	0	$\frac{1}{8}$	$\frac{1}{8}$	0	$\frac{1}{8}$	$\frac{1}{8}$	0	0	0
a	0	0	0	0	0	$\frac{1}{4}$	0	0	0	0	$\frac{1}{4}$	0	0	$\frac{1}{4}$	0	$\frac{1}{4}$
b	0	0	0	0	$\frac{1}{8}$	0	0	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	0	$\frac{1}{4}$	0	$\frac{1}{8}$	0	$\frac{1}{8}$
c	0	0	$\frac{1}{4}$	0	0	$\frac{1}{8}$	$\frac{1}{8}$	0	$\frac{1}{8}$	$\frac{1}{8}$	0	0	$\frac{1}{8}$	0	$\frac{1}{8}$	0
d	0	0	0	$\frac{1}{8}$	0	0	$\frac{1}{8}$	$\frac{1}{4}$	0	0	$\frac{1}{4}$	$\frac{1}{8}$	0	0	$\frac{1}{8}$	0
e	0	$\frac{1}{8}$	0	0	0	0	0	$\frac{1}{8}$	$\frac{1}{8}$	0	0	0	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{8}$
f	0	0	0	$\frac{1}{8}$	0	0	$\frac{1}{8}$	0	0	0	$\frac{1}{4}$	$\frac{1}{8}$	0	0	$\frac{1}{8}$	$\frac{1}{4}$

2.3 迭代型分组密码的 DPT

本节我们研究迭代型分组密码的 DPT 与其轮函数 DPT 之间的关系。为了简单起见, 我们首先研究一种简化后的模型, 即如图 2 所示的 SKS 结构。SKS 结构包含两个相同的 S 盒和一个异或密钥操作, 设输入为 x 、输出为 y , 那么 SKS 结构的表达式可以写成:

$$y = S(S(x) \oplus k)$$

其中, k 表示密钥, 下面我们研究如何通过 S 盒的 DPT 来构造 SKS 结构的 DPT。

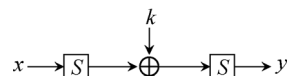


图 2 SKS 结构

Figure 2 The SKS structure

构造 SKS 结构的 DPT 时, 最直观的方法是穷举所有的明文和所有的密钥。这样在构造多轮 SKS 结构的 DPT 时, 由于此时需要穷举的子密钥很多, 构造 DPT 的复杂度会非常大。因此我们考虑使用另一种方法来构造整个 SKS 结构的 DPT。我们用 D_{SKS} 表示 SKS 结构的 DPT 矩阵, D_S 表示单个 S 盒的 DPT 矩阵, 我们研究如何使用 D_S 得到 D_{SKS} 。

令 SKS 结构的输入差分为 α , 输出差分为 γ 。对于第一个 S 盒, 当输入差分为 α 时会有若干个输

出差分, 记第一个 S 盒的输出差分集合为 $\{\beta_i\}$ 。然后我们计算差分特征:

$$\alpha \xrightarrow{\text{SKS}} \gamma$$

的概率, 也就是

$$\alpha \xrightarrow{S} \{\beta_i\} \xrightarrow{S} \gamma$$

的差分概率, 我们有

$$\begin{aligned} \Pr[\alpha \xrightarrow{\text{SKS}} \gamma] &= \sum_i \Pr[\alpha \xrightarrow{S} \beta_i] \times \Pr[\beta_i \xrightarrow{S} \gamma] \\ &= \sum_i D[\alpha, \beta_i] \times D[\beta_i, \gamma] \\ &= D^2[\alpha, \gamma] \end{aligned}$$

也就是说, 对于 SKS 结构, 我们有 $D_{\text{SKS}} = D_S^2$ 。

因此, 当我们得到单个 S 盒的 DPT 矩阵后, 只需要通过一次矩阵乘就能够得到 SKS 结构的 DPT 矩阵。表 5 表示当 S 盒为 CRAFT 的 S 盒时的 SKS 结构的 DPT, 表 5 的结果是基于表 4 做矩阵乘得到的。

表 5 SKS 结构的 DPT

Table 5 The DPT of SKS structure

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	$\frac{5}{32}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{32}$	$\frac{1}{32}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{3}{32}$	$\frac{1}{32}$	$\frac{1}{32}$	$\frac{1}{8}$	$\frac{1}{32}$	$\frac{1}{16}$	$\frac{1}{32}$
2	0	$\frac{1}{16}$	$\frac{1}{4}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{16}$	0	$\frac{1}{8}$	$\frac{1}{16}$	0	$\frac{1}{16}$	$\frac{1}{16}$	0	$\frac{1}{16}$	0
3	0	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{5}{32}$	$\frac{1}{32}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{16}$	$\frac{3}{32}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$
4	0	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{5}{32}$	$\frac{1}{32}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{32}$	$\frac{1}{16}$	$\frac{3}{32}$	$\frac{1}{32}$	$\frac{1}{32}$	$\frac{1}{32}$
5	0	$\frac{1}{32}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{3}{16}$	$\frac{1}{16}$	0	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{16}$
6	0	$\frac{1}{32}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{5}{32}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{16}$	$\frac{5}{32}$	$\frac{1}{16}$
7	0	$\frac{1}{16}$	0	$\frac{1}{16}$	$\frac{1}{16}$	0	$\frac{1}{16}$	$\frac{3}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$
8	0	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{16}$	0	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$
9	0	$\frac{3}{32}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{8}$	$\frac{1}{32}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{5}{32}$	$\frac{1}{32}$	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{32}$	$\frac{1}{32}$	$\frac{1}{32}$
a	0	$\frac{1}{32}$	0	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{8}$	0	$\frac{1}{32}$	$\frac{1}{4}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{8}$
b	0	$\frac{1}{32}$	$\frac{1}{16}$	$\frac{3}{32}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{5}{32}$	$\frac{1}{32}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$
c	0	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{3}{32}$	$\frac{1}{32}$	$\frac{1}{32}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{32}$	$\frac{1}{32}$	$\frac{5}{32}$	$\frac{1}{32}$	$\frac{1}{16}$	$\frac{1}{32}$
d	0	$\frac{1}{32}$	0	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{3}{16}$	$\frac{1}{16}$	$\frac{1}{8}$
e	0	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{16}$	$\frac{3}{32}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{5}{32}$	$\frac{1}{16}$
f	0	$\frac{1}{32}$	0	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{3}{16}$

(注: S 是 CRAFT 的 S 盒)

对于迭代型分组密码, 我们可以将其轮函数 F 看成是一个大 S 盒, 并记其 DPT 矩阵为 D_F 。因此对于 r 轮的迭代型分组密码, 其 DPT 矩阵就是 D_F^r , 这样我们就通过轮函数的 DPT 得到了整个分组密码的 DPT, 并且这个 DPT 中包含了分组密码中的全部差分信息。

需要说明的是, 迭代型分组密码一般使用 SPN 结构, 其轮函数包括一个若干 S 盒并行的 S 层和一个线性层, 我们可以通过 S 层和线性层的性质来比较容易地构造出整个轮函数的 DPT, 而不需要穷举轮函数所有的输入。

2.4 轮函数 DPT 的性质

在本节中, 我们总结迭代型分组密码轮函数的 DPT 矩阵和随机置换的 DPT 矩阵的性质。

设 D 是一个分组长度为 n 比特的迭代型分组密码的 DPT 矩阵, 显然 D 是一个 2^n 阶方阵, 元素 $D[i, j]$ 表示输入差分为 i , 输出差分为 j 的差分概率。 D 满足:

$$\begin{cases} D[0, 0] = 1 \\ D[i, 0] = 0, \text{ 任意 } i \neq 0 \\ D[0, j] = 0, \text{ 任意 } j \neq 0 \\ \sum_{i=0}^{2^n-1} D[i, k] = 1, \text{ 任意 } k \\ \sum_{j=0}^{2^n-1} D[k, j] = 1, \text{ 任意 } k \end{cases}$$

而如果一个分组密码能够完全地抵抗差分攻击, 那么其 DPT 矩阵 R 的形式应该为

$$\begin{cases} R[0, 0] = 1 \\ R[i, 0] = 0, \text{ 任意 } i \neq 0 \\ R[0, j] = 0, \text{ 任意 } j \neq 0 \\ R[i, j] = \frac{1}{2^n - 1}, \text{ 任意 } i, j \neq 0 \end{cases}$$

实际上, 对于任意一个给定的分组密码, 其 DPT 矩阵都不可能完全与 R 相等, 但我们认为随机置换的 DPT 矩阵与 R 相同, 即如果一个分组密码的 DPT 矩阵能够与 R 区分开来, 那么我们就可以使用差分攻击将这个分组密码与随机置换区分开来, 并且我们称随机置换的差分概率分布是均匀的。

3 使用 DPT 证明一些差分性质

本节利用 DPT 矩阵的性质来研究迭代型分组密码的一些差分性质。首先我们说明任意的 r 轮迭代分组密码都不会完全抵抗差分攻击, 即其 DPT 矩阵 D^r 不可能与随机置换的 DPT 矩阵 R 相等。

引理 1. 对任意的 r 轮迭代型分组密码 $E^{(r)}$, 其 DPT 矩阵 D^r 一定满足:

$$D^r \neq R$$

证明. 我们的目标是证明: D^r 的每个矩阵元都不可能等于 $\frac{1}{2^n-1}$. $E^{(r)}$ 的 DDT 矩阵 $\tilde{D}^r$ 与 DPT 矩阵 D^r 之间关系为

$$D^r[i, j] = \frac{\tilde{D}^r[i, j]}{2^n}$$

因为 $\tilde{D}^r$ 的矩阵元均为偶数, 所以 D^r 的矩阵元形式为 $\frac{k}{2^{rn-1}}$, 其中 k 为正整数. 显然 $\frac{k}{2^{rn-1}} \neq \frac{1}{2^n-1}$, 否则有 $k = \frac{2^{rn-1}}{2^n-1}$, 与 k 为整数矛盾.

由引理 1 可知, 任意的 r 轮迭代分组密码的差分概率分布一定不会完全均匀, 即一定有 $D^r \neq R$. 但随着轮数的增大, D^r 可能与 R 非常接近以至于计算上不可区分. 接下来我们研究当轮数足够大时, 分组密码的 DPT 矩阵是否会收敛到 R .

3.1 差分特征概率分布的收敛性

本节使用 DPT 矩阵对差分性质进行一些证明, 这里我们约定我们研究的分组密码的轮函数是一个双射, 且只存在一个概率为 1 的差分特征 $0 \rightarrow 0$, 当提及最大差分概率时, 我们不考虑这种概率为 1 的差分特征, 即只考虑非平凡的差分特征的差分概率及其分布情况.

首先, 我们使用矩阵的性质证明了迭代型分组密码最大差分概率和最小差分概率随轮数的变化情况.

定理 1. 对任意的 r 轮迭代型分组密码 $E^{(r)}$, 设其最大差分概率为 $p_{\max}^r$, 最小差分概率为 $p_{\min}^r$, 那么对于任意的整数 $R > r$, 都有

$$p_{\max}^R \leq p_{\max}^r \text{ 和 } p_{\min}^R \geq p_{\min}^r$$

即随着轮数的增加, 迭代型分组密码的最大差分概率不会变大, 最小差分概率不会变小.

证明. 首先, 我们证明 $p_{\max}^R \leq p_{\max}^r$, 即 $E^{(r)}$ 的最大差分概率不会变大. $E^{(r)}$ 的轮函数的 DPT 矩阵为 D , $E^{(r)}$ 的 DPT 矩阵为 D^r . 我们只需要证明当 $R > r$ 时有 $\max\{D^R\} \leq \max\{D^r\}$ 即可, 设 $\max\{D^R\} = D^R[p, q]$, 那么我们有

$$\begin{aligned} \max\{D^R\} &= \max\{D^{R-r} \times D^r\} \\ &= \sum_{i=0}^{2^n-1} D^{R-r}[p, i] \times D^r[i, q] \\ &\leq \max\{D^r\} \times \sum_{i=0}^{2^n-1} D^{R-r}[p, i] \\ &\leq \max\{D^r\} \end{aligned}$$

因此最大差分概率随轮数增加不增, 同理可证 $p_{\min}^R \geq p_{\min}^r$, 即最小差分概率随轮数增加不降.

由定理 1 可知, 随着轮数的增加, 迭代型分组密码的最大差分概率不会增加, 最小差分概率不会下降. 也就是说, 迭代型分组密码抵抗差分分析的能力随着轮数的增加不会减弱. 下面我们引入马尔可夫矩阵来研究当轮数足够大时, 迭代型分组密码的差分概率分布的收敛性.

如果一个矩阵满足: 每个元素都是非负的; 每列元素相加等于 1. 则称该矩阵为一个马尔可夫矩阵^[12]. 由于我们这里研究的迭代型分组密码的轮函数是一个双射, 又因为 DPT 矩阵中的元素为差分概率, 所以 DPT 矩阵 D 的每行元素之和为 1, 每列元素之和也为 1, 因此 D 是一个马尔可夫矩阵. 那么我们就可以使用马尔可夫矩阵的性质来研究差分性质. 我们设 M 是一个 m 阶的马尔可夫矩阵, $\lambda_0, \lambda_1, \dots, \lambda_{m-1}$ 是它的 m 个特征值(其中可以有重复), 它们对应的特征向量为 $\eta_0, \eta_1, \dots, \eta_{m-1}$, 那么马尔可夫矩阵有如下性质:

性质 1. 对马尔可夫矩阵 M , 至少存在一个 i 使得 $\lambda_i = 1$, 且对任意的 i , 有 $|\lambda_i| \leq 1$.

性质 2. 马尔可夫矩阵 M 的幂仍然是马尔可夫矩阵.

性质 3. 对马尔可夫矩阵 M , 假设其存在 s 个 i 满足 $\lambda_i = 1$, 不妨设为 $\lambda_0 = \lambda_1 = \dots = \lambda_{s-1} = 1$, 且对应的特征向量记为 $\eta_0, \eta_1, \dots, \eta_{s-1}$, 那么对于一个 m 维向量 $\bar{x}$, 以及一个足够大的整数 k , 我们有

$$\lim_{k \rightarrow \infty} M^k \bar{x} = \sum_{i=0}^{s-1} c_i \eta_i$$

其中, $c_i = \frac{\langle \bar{x}, \eta_i \rangle}{|\eta_i|}$.

由性质 3 可知, 如果 M 仅有一个值为 1 的特征值, 不妨设为 $\lambda_1 = 1$, 那么对任意向量 $\bar{x}$, 当 k 足够大时, $M^k \bar{x}$ 一定收敛到 $c_1 \eta_1$; 而如果 M 至少有 2 个值为 1 的特征值, 那么对不同的向量 $\bar{x}$, $M^k \bar{x}$ 可能会收敛到不同的值.

r 轮迭代型分组密码的 DPT 矩阵 D^r 是一个 2^n 阶的马尔可夫矩阵, 由马尔可夫矩阵的性质 3 可以很容易地得到定理 2.

定理 2. 对轮函数是双射的 r 轮迭代型分组密码 $E^{(r)}$, 随着 r 的增长, $E^{(r)}$ 的差分概率分布一定收敛到一个确定状态.

证明. 若 $E^{(r)}$ 的轮函数的 DPT 矩阵 D 有 s 个值为

1 的特征值 $\lambda_0 = \lambda_1 = \dots = \lambda_{s-1} = 1$, 且这些特征值对应的特征向量为 $\eta_0, \eta_1, \dots, \eta_{s-1}$ 。将 D 写成 $D = (\vec{d}_0, \vec{d}_1, \dots, \vec{d}_{2^n-1})$, 其中 $\vec{d}_i$ 为列向量, 这时有

$$\begin{aligned} D^r &= D^{r-1} \cdot D \\ &= D^{r-1} \cdot (\vec{d}_0, \vec{d}_1, \dots, \vec{d}_{2^n-1}) \\ &= (D^{r-1}\vec{d}_0, D^{r-1}\vec{d}_1, \dots, D^{r-1}\vec{d}_{2^n-1}) \end{aligned}$$

根据性质 3, 当轮数 r 趋于无穷时, 有 $D^{r-1}\vec{d}_i$ 收敛到 $\sum_{j=0}^{s-1} c_{i,j} \eta_j$, 其中 $c_{i,j} = \frac{\langle \vec{d}_i, \eta_j \rangle}{|\eta_j|}$ 。因此, D^r 一定收敛到某个确定的矩阵, 即 $E^{(r)}$ 的差分概率分布一定随 r 收敛到一个确定状态。

下面我们研究满足什么条件的 D 能够收敛到 R , 如果 D^r 能够收敛到 R 就说明在足够多的轮数后这个密码算法一定没有高概率差分。

定理 3. 若轮函数是双射的 r 轮迭代型分组密码 $E^{(r)}$ 的轮函数 DPT 矩阵 D 只有两个值为 1 的特征值, 那么当轮数 r 足够大时, $E^{(r)}$ 的差分概率分布一定收敛为均匀分布, 即

$$\lim_{r \rightarrow \infty} D^r = R$$

证明. 根据 2.4 节的讨论, 我们可以将 D 和 R 表示成如下的分块矩阵, 即

$$D = \begin{bmatrix} 1 & \vec{0}^T \\ \vec{0} & A \end{bmatrix}, \quad R = \begin{bmatrix} 1 & \vec{0}^T \\ \vec{0} & B \end{bmatrix}$$

其中, A 和 B 都是 $(2^n - 1)$ 阶的方阵, 且 B 的所有矩阵元均等于 $\frac{1}{2^n - 1}$ 。

由于 A 的每行元素求和与每列元素求和均为 1, 所以 A 和 A^T 都是马尔可夫矩阵。根据马尔可夫矩阵的性质 1, A 存在一个值为 1 的特征值, 且这个特征值对应一个 $2^n - 1$ 维的特征向量 $\eta_1 = (1, 1, \dots, 1)^T$, 因为此时有: $A\eta_1 = \eta_1$ 。如果 A 只有一个值为 1 的特征值, 那么对于任意 $2^n - 1$ 维的向量 $\vec{d}_i$, 我们有

$$\lim_{r \rightarrow \infty} A^r \vec{d}_i = c_i \eta_1$$

因此当 r 足够大时有: $\lim_{r \rightarrow \infty} D^r = (c_1 \eta_1, c_2 \eta_1, \dots, c_{2^n-1} \eta_1)$

又因为 A 中每列所有元素求和均为 1, 所有可以推出对任意 i 都有 $c_i = \frac{1}{2^n - 1}$, 所以此时有

$$\lim_{r \rightarrow \infty} D^r = R$$

由 D 的性质可知 A 的特征值一定是 D 的特征值, 所以此时 D 存在一个值为 1 的特征值, 且其特征向量为 $(0, 1, 1, \dots, 1)^T$ 。另外, 令 $\vec{d}_0 = (1, 0, 0, \dots, 0)^T$, 显然有 $D^r \vec{d}_0 = \vec{d}_0$, 所以 D 还存在另一个值为 1 的特征值, 这个特征值对应的特征向量为 $\vec{d}_0 = (1, 0, 0, \dots, 0)^T$ 。所以 D 至少有两个值为 1 的特征值, 且如果 D 只有两个值为 1 的特征值, 那么一定有

$$\lim_{r \rightarrow \infty} D^r = R$$

定理得证。

由此, 我们说明了: 若 D 只有两个值为 1 的特征值, 那么当 r 足够大时, D^r 一定收敛到 R , 即 r 轮迭代型分组密码的差分概率分布一定收敛为均匀分布。然而, 当 D 有多于两个值为 1 的特征值时, D^r 不一定收敛到 R 。下面我们通过以下两个实例来说明这一事实。

例 1. 我们考虑如图 1 的迭代型分组密码, 并令其轮函数为 4 比特 S 盒, 当这个 S 盒是 PRESENT^[13] 的 S 盒时, 其 DPT 矩阵只有两个值为 1 的特征值, 经过实验我们发现当 r 足够大时其 r 轮的差分概率分布表趋于 R , 因此当轮数足够大时其不会有高概率差分特征。

下面我们通过例 2 来说明: 当轮函数的 DPT 矩阵有多于两个值为 1 的特征值时, r 轮 DPT 矩阵不一定趋于 R 。

例 2. 定义一个新的 S 盒, 其置换为

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

这个 S 盒的 DPT 矩阵有五个值为 1 的特征值, 其 100 轮的 DPT 见表 6, 而且随着轮数增长其 DPT 不会再改变, 显然其存在任意轮的高概率差分特征。

根据定理 3, 对于一个轮函数, 我们可以通过计算其 DPT 矩阵的值为 1 的特征值个数来估计足够高的轮数的分组密码的差分性质。我们也可以使用定理 3 来评价一个轮函数或 S 盒的安全性, 即认为当轮函数或 S 盒的 DPT 矩阵只存在两个值为 1 的特征值时, 密码算法在足够高轮数下一定能够抵抗差分攻击, 而当值为 1 的特征值多于两个时就要仔细检查该密码算法是否具有任意轮的高概率差分闭包。

3.2 对合轮函数的差分性质

本节我们考虑当迭代型分组密码的轮函数是对合轮函数时的分组密码的差分性质。对合轮函数也是双射的轮函数, 对任意输入 x , 对合轮函数 F 满足:

$$F(x) = F^{-1}(x)$$

当迭代型分组密码的轮函数是对合轮函数时, 加密解密过程完全相同, 可以使用同一套线路, 这样大大简化了线路设计, 减少了线路面积。目前很多轻量级分组密码如 CRAFT, MIDORI 等在轮函数中都会使用对合的元件, 如对合 S 盒、对合线性层等。也有一些轻量级密码算法如 Loong^[14], 其每轮的轮函数就是一个对合的轮函数。

表 6 自定义 S 盒 100 轮的 DPT
Table 6 The DPT of self-defined S-box

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	$\frac{1}{4}$	0	0	0	0	$\frac{1}{4}$	0	0	0	0	$\frac{1}{4}$	$\frac{1}{4}$	0	0	0
2	0	0	$\frac{1}{4}$	0	0	$\frac{1}{4}$	0	0	$\frac{1}{4}$	0	0	0	0	0	0	$\frac{1}{4}$
3	0	0	0	$\frac{1}{4}$	$\frac{1}{4}$	0	0	0	0	$\frac{1}{4}$	0	0	0	0	$\frac{1}{4}$	0
4	0	0	0	$\frac{1}{4}$	$\frac{1}{4}$	0	0	0	0	$\frac{1}{4}$	0	0	0	0	$\frac{1}{4}$	0
5	0	0	$\frac{1}{4}$	0	0	$\frac{1}{4}$	0	0	$\frac{1}{4}$	0	0	0	0	0	0	$\frac{1}{4}$
6	0	$\frac{1}{4}$	0	0	0	0	$\frac{1}{4}$	0	0	0	0	$\frac{1}{4}$	$\frac{1}{4}$	0	0	0
7	0	0	0	0	0	0	0	$\frac{1}{2}$	0	0	0	0	0	$\frac{1}{2}$	0	0
8	0	0	$\frac{1}{4}$	0	0	$\frac{1}{4}$	0	0	$\frac{1}{4}$	0	0	0	0	0	0	$\frac{1}{4}$
9	0	0	0	$\frac{1}{4}$	$\frac{1}{4}$	0	0	0	0	$\frac{1}{4}$	0	0	0	0	$\frac{1}{4}$	0
a	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
b	0	$\frac{1}{4}$	0	0	0	0	$\frac{1}{4}$	0	0	0	0	$\frac{1}{4}$	$\frac{1}{4}$	0	0	0
c	0	$\frac{1}{4}$	0	0	0	0	$\frac{1}{4}$	0	0	0	0	$\frac{1}{4}$	$\frac{1}{4}$	0	0	0
d	0	0	0	0	0	0	0	$\frac{1}{2}$	0	0	0	0	0	$\frac{1}{2}$	0	0
e	0	0	0	$\frac{1}{4}$	$\frac{1}{4}$	0	0	0	0	$\frac{1}{4}$	0	0	0	0	$\frac{1}{4}$	0
f	0	0	$\frac{1}{4}$	0	0	$\frac{1}{4}$	0	0	$\frac{1}{4}$	0	0	0	0	0	0	$\frac{1}{4}$

对合轮函数的 DPT 矩阵 D 一定是一个对称矩阵, 因为有

$$\Pr[\alpha \xrightarrow{F} \beta] = \Pr[\beta \xrightarrow{F^{-1}} \alpha] = \Pr[\beta \xrightarrow{F} \alpha]$$

使用对称矩阵的一些性质, 我们可以对有着对合轮函数的迭代型分组密码做进一步的分析。我们先研究一个比较简单的 SKS 模型, 其中 S 为对合的 S 盒, 我们说明 SKS 结构的最大差分概率(或之一)一定位于其 DPT 矩阵的主对角线上。

定理 4. 对 S 盒有对合性质的 SKS 结构, 设其最大差分概率为 $p_{\max}$, 则一定存在一个差分 α , 满足:

$$\Pr[\alpha \xrightarrow{\text{SKS}} \alpha] = p_{\max}$$

证明. 记 SKS 结构的 DPT 矩阵为 D_{SKS} , 并设其最大差分概率为 $p_{\max}$, 我们的目标是证明存在一个 α 满足 $D_{\text{SKS}}[\alpha, \alpha] = p_{\max}$ 。

S 盒的 DPT 矩阵 D 为

$$D = \begin{bmatrix} 1 & \vec{0}^T \\ \vec{0} & A \end{bmatrix}$$

矩阵 A 就完全表示了这个 S 盒的所有非平凡差分特征的概率分布情况。由于 S 盒有对合性质, 所以 D 和 A 都是对称矩阵。下面我们只需要证明必然存在一个 A^2 中的最大值位于其主对角线上即可。

设 $A = (\vec{a}_1, \dots, \vec{a}_{2^n-1})^T$, 其中 $\vec{a}_i$ 为行向量, 由于 A 是对称阵, 所以我们有

$$A^2[s, t] = \langle \vec{a}_s, \vec{a}_t \rangle$$

其中, 符号 $\langle \cdot, \cdot \rangle$ 表示两向量的内积。接下来我们只需要证明对于任意的 $s \neq t$, 都有 $A^2[s, t] \leq A^2[s, s]$ 或 $A^2[s, t] \leq A^2[t, t]$ 。

令 $\alpha = \vec{a}_s, \beta = \vec{a}_t$, 现在需要证明:

$$\langle \alpha, \beta \rangle \leq \langle \alpha, \alpha \rangle \text{ 或 } \langle \alpha, \beta \rangle \leq \langle \beta, \beta \rangle$$

使用反证法, 假设有

$$\langle \alpha, \beta \rangle > \langle \alpha, \alpha \rangle \text{ 且 } \langle \alpha, \beta \rangle > \langle \beta, \beta \rangle$$

那么有

$$\langle \alpha, \beta \rangle^2 > \langle \alpha, \alpha \rangle \times \langle \beta, \beta \rangle$$

而根据柯西不等式, 有

$$\langle \alpha, \beta \rangle^2 \leq \langle \alpha, \alpha \rangle \times \langle \beta, \beta \rangle$$

两者矛盾, 因此有 $A^2[s, t] \leq A^2[s, s]$ 或 $A^2[s, t] \leq A^2[t, t]$, 所以矩阵 A^2 的最大矩阵元(或之一)一定位于其主对角线上, 即 SKS 结构一定存在形如 $\alpha \xrightarrow{\text{SKS}} \alpha$ 的差分特征对应着最大差分概率。

根据定理 4, 我们很容易得到如下的推论。

推论 1. 对一个 r 轮迭代型分组密码, 其中, $r = 2^k$ (k 为正整数), 若其轮函数是对合的, 设其最大差分概率为 $p_{\max}$, 则一定存在一个差分 α , 满足:

$$\Pr[\alpha \xrightarrow{\text{SKS}} \alpha] = p_{\max}$$

推论 1 为分析有着对合轮函数的迭代型分组密码的差分性质提供了帮助, 在寻找其高概率差分特征时, 就只需找形如 $\alpha \xrightarrow{r} \alpha$ 的差分特征即可。这大大减少了寻找高概率差分特征的复杂度。进一步地, 我们说明当轮数为 $r = 2^k$ 时, SKS 结构的所有形如 $\alpha \xrightarrow{\text{SKS}} \alpha$ 的差分特征的差分概率都大于 $\frac{1}{2^n-1}$ 。

定理 5. 对于 S 盒为 n 比特对合 S 盒的 SKS 结构, 所有形如 $\alpha \xrightarrow{\text{SKS}} \alpha$ 的差分特征的差分概率都大于

$\frac{1}{2^n-1}$, 即

$$\Pr[\alpha \xrightarrow{\text{SKS}} \alpha] > \frac{1}{2^n-1}$$

证明. 设 S 盒的 DPT 矩阵为 $\mathbf{D}$, SKS 结构的 DPT 矩阵为 $\mathbf{D}_{\text{SKS}}$, 将 $\mathbf{D}$ 用行向量表示为

$$\mathbf{D} = (\vec{d}_0, \vec{d}_1, \dots, \vec{d}_{2^n-1})^T$$

我们只需要证明 $\mathbf{D}_{\text{SKS}}$ 主对角线上的所有元素都大于 $\frac{1}{2^n-1}$ 即可. 因为 $\mathbf{D}_{\text{SKS}} = \mathbf{D}^2$, 所以 $\mathbf{D}_{\text{SKS}}$ 中的主对角线元素为

$$\mathbf{D}_{\text{SKS}}[i, i] = \langle \vec{d}_i, \vec{d}_i \rangle$$

其中, $\langle \cdot, \cdot \rangle$ 表示内积, 向量 $\vec{d}_i$ 满足:

$$\begin{cases} \vec{d}_0 = (1, 0, 0, \dots, 0), i = 0 \\ \vec{d}_i = (0, d_{i,1}, d_{i,2}, \dots, d_{i,2^n-1}), \square i \neq 0 \\ \sum_j d_{i,j} = 1, \square \text{任意 } i \end{cases}$$

因此有

$$\langle \vec{d}_i, \vec{d}_i \rangle = \sum_j d_{i,j}^2 \geq \left(\frac{1}{2^n-1}\right)^2 \times (2^n-1) \geq \frac{1}{2^n-1}$$

等号成立的条件为

$$\vec{d}_i = (0, \frac{1}{2^n-1}, \frac{1}{2^n-1}, \dots, \frac{1}{2^n-1})$$

由引理 1 知, 等号无法成立, 定理得证。

虽然对任意 α 都有 $\Pr[\alpha \xrightarrow{\text{SKS}} \alpha] > \frac{1}{2^n-1}$, 但

实际上这个概率可能与 $\frac{1}{2^n-1}$ 非常接近, 以至于对于单条差分特征, 我们无法区分 SKS 结构与随机置换. 但在实际攻击中我们可以不只计算一条差分特征的概率, 而是计算多个形式为 $\alpha \rightarrow \alpha$ 的差分概率并求和, 这样可能会得到有效的区分器。

在本章的最后, 对于有对合轮函数的迭代型分组密码, 我们研究其不同轮之间的最大差分概率之间的关系。

定理 6. 对于轮函数有对合性质的迭代型分组密码 E , 其 $r_k = 2^k$ 轮的最大差分概率 m_k 与 $r_{k+1} = 2^{k+1}$ 轮的最大差分概率 m_{k+1} 关系为

$$m_k \geq m_{k+1} > m_k^2 + \frac{(1-m_k)^2}{2^n-2}$$

证明. 由定理 1 显然有 $m_k \geq m_{k+1}$, 下面证明

$$m_{k+1} > m_k^2 + \frac{(1-m_k)^2}{2^n-2}.$$

设 $r_k = 2^k$ 轮 $E^{(r_k)}$ 的 DPT 矩阵为

$\mathbf{D} = (\vec{d}_0, \vec{d}_1, \dots, \vec{d}_{2^n-1})^T$, 其中 $\vec{d}_i, i \in \{0, \dots, 2^n-1\}$ 是行向量. 由定理 4 知, 存在 i 使得 $r_{k+1} = 2^{k+1}$ 轮的最大差分概率为 $m_{k+1} = \langle \vec{d}_i, \vec{d}_i \rangle$, 其中, $\vec{d}_i = (0, d_{i,1}, d_{i,2}, \dots, d_{i,2^n-1})$, 因此有

$$\begin{aligned} m_{k+1} &\geq m_k^2 + \left(\frac{1-m_k}{2^n-2}\right)^2 \times (2^n-2) \\ &= m_k^2 + \frac{(1-m_k)^2}{2^n-2} \end{aligned}$$

等号成立的条件为

$$\vec{d}_i = \left(0, \frac{1-m_k}{2^n-2}, \frac{1-m_k}{2^n-2}, \dots, m_k, \dots, \frac{1-m_k}{2^n-2}\right)$$

下面证明等式成立的条件不能满足. 使用与引理 1 类似的方法, 我们假设有 $\frac{1-m_k}{2^n-2} = \frac{2k}{2^n}$, 其中 k 为正整数, 那么有

$$k = \frac{2^{n-1}(1-m_k)}{2^n-2}$$

当 $n \geq 2$ 时有 $k < 1$, 与 k 为正整数矛盾, 所以等式成立的条件无法满足, 定理得证。

需要说明的是, 在定理 6 的证明中我们使用放缩法, 得到的实际上是一个某个轮数下最大差分概率的下界, 而且这个下界随轮数上升而下降得非常快, 也就是如果只使用定理 6 中的不等式对差分概率进行估计, 那么经过较少的轮数我们估计的 m_k 就会很快地接近 $\frac{1}{2^n-1}$. 但对于实际的密码算法, 其很有可能达不到我们证明中的放缩的下界, 所以其最大差分概率不一定会很快地接近 $\frac{1}{2^n-1}$ 。

4 字节级截断差分

由于实际使用中的分组密码的分组长度较长, 以现有的计算机的储存和计算能力, 我们无法构造和存储整个分组密码的 DPT. 因此在本节研究迭代型分组密码的字节级截断差分^[15-16]. 多数迭代型分组密码如 AES, CRAFT 等都是基于字节或半字节操作的, 字节级截断差分是指我们只研究每个 cell 或字节上是否有差分, 而不考虑差分的值, 这样我们能以一个可接受的复杂度构造整个迭代型分组密码的字节级截断差分的差分概率分布表 TPT. 这个表中包含了迭代型分组密码中全部的字节级截断差分信息。

4.1 迭代型分组密码 TPT 构造

本节通过对图 3 所示的 Toy Cipher 构造 TPT 的过程来说明如何构造 SPN 结构的迭代型分组密码的 TPT, 并介绍了随机置换的 TPT 矩阵的形式。

Toy Cipher 是一个分组长度为 16 比特的 SPN 结

构的迭代型分组密码, 其每轮的状态分成 4 个 4-bit 的 cell 进行操作, 如图 3。每轮包括 4 个操作: 列混淆(MC), 字节置换(PN), S 盒(SB)和异或密钥(AK)。第 t 轮中每个操作的输入状态记为 x_t, y_t, z_t, w_t , 并用 $x_t[i]$ 表示状态中的第 i 个 cell, 每轮的 4 个操作具体如下。

(1) 列混淆操作是对状态左乘一个矩阵, 即

$$\begin{pmatrix} y_t[0] \\ y_t[1] \\ y_t[2] \\ y_t[3] \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_t[0] \\ x_t[1] \\ x_t[2] \\ x_t[3] \end{pmatrix} = \begin{pmatrix} x_t[0] \oplus x_t[2] \oplus x_t[3] \\ x_t[1] \oplus x_t[3] \\ x_t[2] \\ x_t[3] \end{pmatrix}$$

(2) 字节置换操作为

$$(z_t[0], z_t[1], z_t[2], z_t[3]) = (y_t[3], y_t[2], y_t[1], y_t[0])$$

(3) S 盒操作是将 z_t 中的每个 cell 并行地经过 S 盒, 这里 S 盒使用 CRAFT 算法的 S 盒, 即

$$w_t[i] = S(z_t[i]), \text{ for } i=0,1,2,3$$

(4) $w_t[i]$ 经过一个密钥异或操作得到下一轮的初始状态 $x_{t+1}[i]$ 。这里我们认为每轮的子密钥都是独立随机的, 即不考虑密钥对差分的影响。

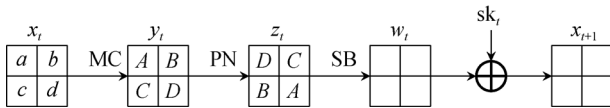


图 3 16 比特 Toy Cipher
Figure 3 16 bit Toy Cipher

现在我们考虑 Toy Cipher 的字节级截断差分。对每个 cell, 用二进制符号‘0’来表示这个 cell 上没有差分, 用二进制符号‘1’来表示这个 cell 上有差分。那么对于 Toy Cipher, 就可以用一个 4 比特的二进制数, 或一个 0 到 15 的十进制整数来表示一个字节级截断差分状态。例如, 设某个状态的差分值为‘0x001a’, 那么我们可以用二进制串‘0011’或整数‘3’来表示这个字节级截断差分状态。进一步地, 我们也可以用这种方法表示一个截断差分路径。例如, 图 4 中的字节级截断差分路径为‘000* → *0***’, 其中符号‘*’表示非零差分, 图中灰色块表示这个 cell 上有差分, 白色块表示这个 cell 上无差分。该字节级截断差分的二进制表示为‘0001 → 1011’, 用十进制整数表示为 1 → 11。下文中我们都只使用十进制整数表示字节级截断差分。我们容易得到图 4 所示的字节级截断差分 1 → 11 的概率为 1。

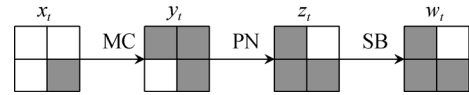


图 4 字节级截断差分 0001 → 1011

Figure 4 Byte-oriented truncated differential Path1

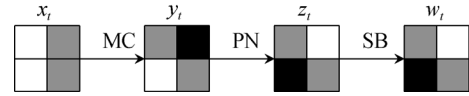


图 5 字节级截断差分 0101 → 10?1

Figure 5 Byte-oriented truncated differential Path2

下面我们考虑另外一条概率小于 1 的字节级截断差分传播路径, 如图 5 所示, 输入截断差分模式为‘0*0*’, 黑色 cell 表示该 cell 上的差分值是随机的, 即不确定该 cell 上是否有差分, 在图 5 中, 我们用‘?’来表示这种状态。由于每个 cell 的大小为 4 比特, 所以黑色 cell 无差分的概率为 2^{-4} , 有差分的概率为 $1 - 2^{-4}$, 这样我们就可以得到图 5 所示的字节级截断差分概率为

$$\Pr[5 \rightarrow 9] = 2^{-4}, \Pr[5 \rightarrow 11] = 1 - 2^{-4}$$

使用相同的方法, 我们可以穷举全部 16 个输入的字级截断差分, 然后计算相应的一轮后的输出字节级截断差分及其概率, 由此我们就能够构造一个字级截断差分概率分布表 T, 如表 7 所示。表

表 7 Toy Cipher 的 TPT
Table 7 The TPT of Toy Cipher

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
2	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\frac{1}{16}$	$\frac{15}{16}$
4	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	$\frac{1}{16}$	0	$\frac{15}{16}$	0	0	0	0
6	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	$\frac{1}{256}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{223}{256}$
8	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	$\frac{1}{16}$	$\frac{15}{16}$	0	0	0	0
a	0	0	0	0	$\frac{1}{16}$	$\frac{15}{16}$	0	0	0	0	0	0	0	0	0	0
b	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\frac{1}{16}$	$\frac{15}{16}$
c	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
d	0	0	0	0	0	0	0	0	$\frac{1}{256}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{223}{256}$	0	0	0	0
e	0	0	0	0	0	0	$\frac{1}{16}$	$\frac{15}{16}$	0	0	0	0	0	0	0	0
f	0	0	0	0	0	0	0	0	0	0	0	0	$\frac{1}{256}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{223}{256}$

中元素 $\Pi[i, j]$ 表示输入截断差分模式为 i , 输出截断差分模式为 j 的概率。

在得到一轮的 TPT 矩阵后, 就能使用矩阵乘的方法得到 r 轮的 TPT 矩阵 T^r 。我们就可以从 T^r 中得到 r 轮迭代型分组密码的全部字节级截断差分信息。

注意到与迭代型分组密码的 DPT 矩阵不同, 迭代型分组密码的 TPT 矩阵只满足每行求和为 1, 而并不要求每列求和为 1, 显然, T 并不是一个马尔可夫矩阵。我们认为随机置换能够完全抵抗字节级截断差分分析, 随机置换的 TPT 矩阵 RT (在这里我们以 16-bit 的随机置换为例) 的元素需满足以下条件。

(1) 当 $i=0$ 时, RT 第一行元素为

$$RT[0, j] = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], i=0$$

(2) 当 $i \neq 0$ 时, 由于对于随机置换来说其输入差分与输出差分无关, 故 RT 每行的元素都相同。对于随机置换, 其输出差分的每个 cell 的截断差分为 0 的概率为 $1/16$, 截断差分不为 0 的概率为 $15/16$, 据此我们可以计算随机置换的输出截断差分的概率。需要注意的是, 当输入截断差分非 0 时, 输出截断差分一定非 0。我们将列索引 j 表示成 4 比特的二进制数, 并设其汉明重量为 $d(j)$ 。容易得到, 当输入截断差分为 $i(i \neq 0)$ 时, 输出截断差分为 $j(j \neq 0)$ 的概率为

$$RT[i, j] = \frac{1}{1-2^{-16}} \times \left(\frac{1}{16}\right)^{4-d(j)} \times \left(\frac{15}{16}\right)^{d(j)}, i \neq 0, j \neq 0$$

此外, 当 $i \neq 0, j=0$ 时, 其输出截断差分为 0 的概率为 0。我们用符号 $RT[i]$ 表示 RT 的第 i 行 ($i \in \{0, 1, \dots, 15\}$) 16 个元素组成的行向量。当 $i \neq 0$ 时, $RT[i]$ 需满足:

$$RT[i] = [0, 2^{-12.09}, 2^{-12.09}, 2^{-8.18}, 2^{-12.09}, 2^{-8.18}, 2^{-8.18}, 2^{-4.27}, 2^{-12.09}, 2^{-8.18}, 2^{-8.18}, 2^{-4.24}, 2^{-8.18}, 2^{-4.27}, 2^{-4.27}, 2^{-0.37}]$$

, 对任意 $i \neq 0$

在对迭代型分组密码进行差分分析时, 如果 T^r 和 RT 有明显差异, 那么我们就构造一个字节级截断差分区分离器。

4.2 针对 Toy Cipher 的实验验证

本节对 Toy Cipher 进行实验以验证我们构造的字节级截断差分布表的正确性。

一方面, 对 5 轮的 Toy Cipher, 我们进行了 4 次实验。每次实验都随机选择输入截断差分模式为‘1’的 10000 个明文对, 令每轮的子密钥都是独立随机的, 计算 5 轮后的输出截断差分模式。最后将 4 次实

验的输出字节级截断差分布情况统计在表 8 中。通过对表 8 得到的实验结果进行简单计算, 我们容易得到输入截断差分模式为‘1’时, 输出截断差分的概率分布情况, 如表 9 所示。另一方面, 我们根据 4.1 节所示的 TPT 的构造方法从理论上构造了 5 轮 Toy Cipher 的 TPT, 即 T^5 , 表 T^5 的第 1 行元素(下标从 0 开始)列在表 10 中。通过表 9 和表 10 的对比, 我们容易观察到由 5 轮 Toy Cipher 的 TPT 计算得到的理论字节级截断差分概率分布与通过真实加密实验得到的字节级截断差分概率分布较为符合。

表 8 5 轮 Toy Cipher 的 4×10000 对明文对实验结果
Table 8 Experiment result of 5-round Toy Cipher with 4×10000 plaintext pairs

j	0	1	2	3	4	5	6	7
1st	0	3	0	29	3	39	21	478
2nd	0	4	0	44	3	54	33	489
3rd	0	3	0	34	3	53	24	493
4th	0	2	0	37	3	52	36	526
sum	0	12	0	144	12	198	114	1986
j	8	9	a	b	c	d	e	f
1st	0	40	37	500	31	547	505	7766
2nd	3	36	42	550	45	520	528	7648
3rd	0	41	29	550	48	510	503	7708
4th	2	51	31	494	40	549	513	7663
sum	5	168	139	2094	164	2126	2049	30785

表 9 5 轮 Toy Cipher 的差分概率分布实验结果
Table 9 Experiment result of 5-round Toy Cipher's differential probability distribution

j	0	1	2	3	4	5	6	7
p	0	$2^{-11.7}$	0	$2^{-8.12}$	$2^{-11.7}$	$2^{-7.66}$	$2^{-8.45}$	$2^{-4.33}$
j	8	9	a	b	c	d	e	f
p	$2^{-12.97}$	$2^{-7.9}$	$2^{-8.17}$	$2^{-4.26}$	$2^{-7.93}$	$2^{-4.23}$	$2^{-4.29}$	$2^{-0.38}$

(注: p 表示字节级截断差分概率)

表 10 5 轮 Toy Cipher 的差分概率分布理论结果
Table 10 Theoretical result of 5-round Toy Cipher's differential probability distribution

j	0	1	2	3	4	5	6	7
p	0	$2^{-12.09}$	0	$2^{-8.09}$	$2^{-12.09}$	$2^{-8.19}$	$2^{-8.19}$	$2^{-4.19}$
j	8	9	a	b	c	d	e	f
p	$2^{-12.19}$	$2^{-8.19}$	$2^{-8.1}$	$2^{-4.29}$	$2^{-8.28}$	$2^{-4.28}$	$2^{-4.19}$	$2^{-0.38}$

(注: p 表示字节级截断差分概率)

根据 4.1 节对随机置换的 TPT 矩阵 RT 的性质分析, 我们容易得到输入截断差分模式为‘1’时, 随机置换的输出字节级截断差分的概率分布, 如表 11

所示。通过表 10 和表 11 的对比, 我们容易观察到 5 轮 Toy Cipher 的 TPT 矩阵 T^5 与随机置换的 TPT 矩阵 RT 有着明显差异。比如, $T^5[1,2]=0$, 而 $RT[1,2]=2^{-12.09}$, 即 Toy Cipher 存在一个字节级截断不可能差分 $1 \xrightarrow{5\text{-rounds}} 2$ 。通过计算我们发现 Toy Cipher 至多有 8 轮的字节级截断不可能差分。随着轮数 r 的增加, T^r 与 RT 越来越接近, 这意味着随着轮数的增加 Toy Cipher 抵抗字节级截断差分分析的能力不断提高。以 100 轮 Toy Cipher 为例, 我们计算其 TPT 矩阵 T^{100} , 并记录 $T^{100}[1, j]$ 于表 12 中。总体来说, $T^{100}[1, j]$ 和 $RT[1, j]$ 相差很小, 但仍然有一些细微差别, 如 $T^{100}[1, 7] \approx 2^{-4.19}$ 但 $RT[1, 7] \approx 2^{-4.27}$ 。另外, 在轮数 r 继续增加的时候, Toy Cipher 的 TPT T^r 几乎不变, 这意味着对于任意轮的 Toy Cipher, 理论上我们都能将其与随机置换区分开来, 虽然这个区分程度可能会非常小以至于不能应用到实际的攻击上。

表 11 $RT[1, j]$ 的所有元素Table 11 All elements of $RT[1, j]$

j	0	1	2	3	4	5	6	7
p	0	$2^{-12.09}$	$2^{-12.09}$	$2^{-8.18}$	$2^{-12.09}$	$2^{-8.18}$	$2^{-8.18}$	$2^{-4.27}$
j	8	9	a	b	c	d	e	f
p	$2^{-12.09}$	$2^{-8.18}$	$2^{-8.18}$	$2^{-4.24}$	$2^{-8.18}$	$2^{-4.27}$	$2^{-4.27}$	$2^{-0.37}$

(注: p 表示字节级截断差分概率)表 12 $J^{100}[1, j]$ 的所有元素Table 12 All elements of $J^{100}[1, j]$

j	0	1	2	3	4	5	6	7
p	0	$2^{-12.28}$	$2^{-12.19}$	$2^{-8.28}$	$2^{-12.19}$	$2^{-8.19}$	$2^{-8.19}$	$2^{-4.19}$
j	8	9	a	b	c	d	e	f
p	$2^{-12.28}$	$2^{-8.19}$	$2^{-8.19}$	$2^{-4.28}$	$2^{-8.28}$	$2^{-4.28}$	$2^{-4.19}$	$2^{-0.38}$

(注: p 表示字节级截断差分概率)

4.3 应用到实际密码算法分析中的复杂度

本节我们研究使用字节级截断差分概率分布表对实际的迭代型分组密码进行分析时的复杂度。

对于多数 SPN 结构的轻量级迭代型分组密码算法, 如 SKINNY^[17], CRAFT, MIDORI 等, 64 比特的明文首先被分成 16 个 4 比特的 cell, 然后再基于 cell 进行多个相同或相似的轮函数操作。轮函数内部的操作一般包括 SB, MC, PN, AK 等操作, 其中 SB 为字节(或 cell)替换, MC 为列混淆, PN 为字节(或 cell)置

换, AK 为异或子密钥。大多数的轮函数都是这些类型操作的组合, 不失一般性, 这种类型的分组密码的轮函数可以表示成如图 6 的形式。

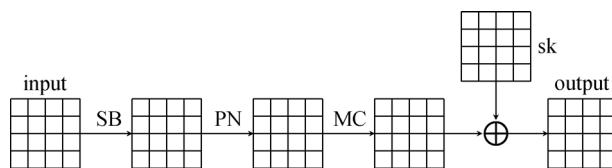


图 6 一般的 SPN 结构

Figure 6 General SPN structure

对于一般的 SPN 结构的迭代型分组密码, 为了寻找字节级截断差分区分器, 我们可以使用类似 4.1 节的方法来计算它的 TPT 矩阵。对于实际的分组密码算法, 一般有 16 个 cell, 因此它的 TPT 矩阵 T 的阶数为 2^{16} , 储存整个表需要的空间复杂度为 2^{32} 。矩阵乘法计算的复杂度为 $O(n^{2.37})$ ^[18], 其中 n 为矩阵的阶数。因此, 我们通过矩阵乘法得到 T^2 的时间复杂度就是 $2^{37.92}$ 。进一步地, 要计算 T^r 时, 可以使用类似 RSA 中的快速模幂算法^[19], 这样只需要 $O(\log_2 r)$ 次矩阵乘法就能从 T 得到 T^r 。此外, 在计算矩阵乘时我们需要预先存储矩阵。因此, 计算 r 轮分组密码的 TPT 矩阵 T^r 的时间复杂度是 $c_1 \cdot 2^{37.92}$, 空间复杂度是 $c_2 \cdot 2^{32}$, 其中 $c_1, c_2 \in O(\log_2 r)$ 。需要注意的是, 在一般的密码分析中, 攻击时间复杂度的衡量标准一般是加密次数。而本文的攻击时间攻击复杂度的衡量标准是乘法次数。本文中使用的矩阵规模较大, 乘法计算的复杂度较高, 需要借助高性能计算机才能在可接受的时间内得到计算结果。为了计算大规模矩阵的乘法, 可以使用 Cannon 算法^[20]来进行并行计算。为了计算两个 N 阶方阵的乘法, Cannon 算法的基本思想是将两个 N 阶方阵各自分成 p 个 $\frac{N}{p}$ 阶的小方阵, 并将它们分配给 $\sqrt{p} \times \sqrt{p}$ 个处理器, 然后处理器进行并行运算, 最终得到两个矩阵的乘积。在使用 p 个处理器的情况下, 计算两个 N 阶方阵的乘法需要的时间复杂度为 $O\left(\frac{N^3}{p}\right)$, 空间复杂度为 $O(N^2)$ 。那么, 对于分组长度为 n 比特、状态分成 $\frac{n}{s}$ 个 cell 且每个 cell 为 s 比特的分组密码算法, 我们计算得到其 TPT 矩阵 T^r 的时间复杂度就为 $O\left(\log_2 r \cdot \frac{2^{\frac{n}{s}}}{p}\right)$, 空间复杂度为 $O\left(\log_2 r \cdot 2^{2\left(\frac{n}{s}\right)}\right)$ 。这样通过计算矩阵乘法就能得到整个密码算法的全部

字节级截断差分性质, 包括一些字节级截断低概率差分 and 不可能差分。

4.4 与基于 MILP 方法搜索字节级截断差分的复杂度对比

目前搜索字节级截断差分时一般使用基于 MILP 方法的搜索方法, 其基本思想是使用一系列的不等式约束来描述分组密码算法, 然后再设定一个合适的目标函数, 将搜索字节级截断差分的问题转化成一个最优化问题, 再使用专门的求解器对这个问题进行求解, 通过得到的解就能得到分组密码的高概率差分特征, 进一步地可以计算其差分概率。本节我们将对比基于 MILP 的方法和基于 TPT 矩阵乘法计算的方法来搜索字节级截断差分的复杂度, 并说明这两种方法的适用范围。

首先我们使用基于 MILP 的方法来搜索 4.2 节中构造的 Toy Cipher 的高概率字节级截断差分。对于不同的轮数, 我们记录生成的不等式约束数量和变量数, 以及求解所用的时间, 见表 13。

表 13 对 Toy Cipher 基于 MILP 方法搜索字节级截断差分的实验结果

Table 13 Experimental results of searching byte level truncated difference for Toy Cipher based on MILP method

轮数	不等式数量	变量数	求解时间
30	1081	496	4.5 秒
40	1441	656	36.15 秒
50	1801	816	73 分钟
60	2161	976	24 小时

从表 13 中可以看到, 不等式数量和变量数量随着轮数线性增加, 但求解时间却上升得非常快。这是因为 MILP 问题是一个 NP 难的问题, 求解的复杂度很高。在使用基于 MILP 方法的自动化密码分析技术对大量算法的分析中, 人们发现 MILP 问题的求解复杂度随着不等式和变量数量的上升而增加得非常快, 当轮数较高时就不能在可接受的时间内得到结果。另外, 其求解时间也很难预测, 对于轮数较高的算法, 我们往往不能估计需要多长时间才能得到结果, 而且实际的计算时间的不确定性很大, 可能是几个星期, 几个月甚至更长, 这就使得基于 MILP 的方法不适合寻找轮数较高的差分。

本文中提出的基于 TPT 矩阵乘法的研究字节级截断差分的方法的主要的时间复杂度来源于矩阵乘法, 其复杂度由 TPT 矩阵的阶数而不是密码算法的轮数决定。以 Toy Cipher 为例, 对于一万轮的 Toy

Cipher 我们能在 0.02 秒之内计算得到其 TPT 矩阵 T^{10000} , 该矩阵包含了一万轮的 Toy Cipher 的全部字节级截断差分特征。对 Toy Cipher 的计算速度如此快的原因是 Toy Cipher 的分组长度很小, 其 TPT 矩阵的阶数只是 16。对于实际的轻量级分组密码算法来说, 其 TPT 矩阵的阶数一般为 2^{16} , 因此矩阵乘法计算的时间会长很多。本文中提出的基于 TPT 矩阵乘法计算来研究字节级截断差分的方法的复杂度为 $\log_2 r \cdot O(n^{2.37})$, 其中 n 为 TPT 矩阵的阶数, r 为算法轮数。当阶数 n 为 2^{16} 时, 计算矩阵乘法的时间复杂度为 $2^{37.92}$ 次乘法, 空间复杂度为 2^{32} 。对于高性能计算机来说, 可以在可接受的时间内得到计算结果, 并且当给定算法轮数时, 我们可以估计得到 r 轮 TPT 矩阵所需要的时间, 且这个计算时间不会随着轮数有较快的增加。这就使得当轮数较高时使用本文中的基于矩阵乘法计算的方法来研究字节级截断差分更为有利。

基于矩阵乘法计算来研究字节级截断差分的另一个优势是, 得到 r 轮的 TPT 矩阵后, 我们可以从这个矩阵中得到这个算法的 r 轮的所有字节级截断差分信息, 包括所有的字节级截断差分的准确差分概率。而使用 MILP 方法一般只能得到一个或几个高概率差分, 并且计算得到的概率也只是实际差分概率的下界。

通过对两种搜索字节级截断差分的方法的对比, 我们可以得到以下结论。在轮数较高且需要准确的字节级截断差分概率时, 本文中提出的方法相比于 MILP 的方法更有优势。在轮数不高且轮函数相对简单时, 基于 MILP 的搜索字节级截断差分的方法可以比较快地找到高概率差分特征, 所以当轮数不高时使用 MILP 方法可以快速找到字节级截断差分特征并能计算其差分概率的一个下界。但随着轮数的增加, MILP 方法的搜索时间会很快地增加, 而且其只能找到一个或几个高概率差分, 而不是差分闭包, 计算得到的差分概率也只是实际差分概率的一个下界。相比于 MILP 方法, 在本文提出的方法中, 虽然计算大规模矩阵乘法的复杂度较高, 但仍然是可以接受的, 而且这个复杂度受轮数变化的影响不大。因此轮数较高且轮函数较为复杂时, 我们的方法更具有优势。此外本文中提出的搜索字节级截断差分的方法可以得到所有的截断差分, 包括字节级截断低概率差分 and 不可能差分, 并且得到的差分概率是准确的。

总结来说, 相比于目前流行的基于 MILP 方法搜

索字节级截断差分的方法来说, 本文中提出的基于 TPT 矩阵乘法计算的方法存在着一定的优势。虽然, 大规模矩阵的存储和乘法计算使得我们暂时无法使用 PC 实现对实际密码算法的攻击。但是, 进一步优化存储方案和矩阵乘的算法可能会为我们的研究带来转机。现阶段, 可以借助高性能计算机来实现实际的攻击。此外, 我们给出的方法的复杂度不会随轮数有较大增长, 并能在可预测的时间内借助 TPT 矩阵得到分组密码的全部准确的字节级截断差分概率, 而不是只能得到差分概率的下界。

5 结论

本文从一个新的角度来研究差分分析, 即使用 DPT 来分析一个轮函数是双射的迭代型分组密码的差分性质, 将迭代分组密码轮函数的 DPT 看成是一个矩阵, 然后利用矩阵和马尔可夫矩阵的性质来分析整个迭代型分组密码的差分性质。我们的主要结论包括对差分概率随轮数的收敛性分析, 对对合轮函数差分性质的分析以及对字节级截断差分概率的分析。我们发现对于一般的轮函数, 如果其 DPT 矩阵只有两个值为 1 的特征值, 那么经过足够高的轮数后一定不存在高概率差分特征。对于对合轮函数, 我们认为它一定存在很多高概率差分特征, 虽然这些差分特征的概率可能非常接近随机置换的差分概率。最后, 我们给出了基于 TPT 矩阵乘法计算来研究字节级截断差分概率的方法。我们以 Toy Cipher 为例, 一方面通过实验验证说明了我们方法的正确性; 另一方面, 给出了基于 MILP 的方法和基于 TPT 矩阵乘法计算的方法来搜索其字节级截断差分的复杂度对比, 并对比了两种方法的优劣和各自的适用场景。在轮数不高且轮函数相对简单时, 基于 MILP 的方法更具有优势。而当密码算法轮数较高时或需要得到准确的差分概率时, 使用本文提出的基于 TPT 矩阵乘法的方法更具有优势。

参考文献

- [1] Biham E, Shamir A. Differential Cryptanalysis of DES Variants[M]. Differential Cryptanalysis of the Data Encryption Standard. New York: Springer, 1993: 33-77.
- [2] Beyne T. Block Cipher Invariants as Eigenvectors of Correlation Matrices[C]. *International Conference on the Theory and Application of Cryptology and Information Security*. Cham: Springer, 2018: 3-31.
- [3] Kim J, Hong S, Lim J. Impossible Differential Cryptanalysis Using Matrix Method[J]. *Discrete Mathematics*, 2010, 310(5): 988-1002.
- [4] Daemen J, Rijmen V. The Design of Rijndael: The Advanced Encryption Standard (AES)[M]. Berlin, Heidelberg: Springer Berlin Heidelberg, 2020.
- [5] Beierle C, Leander G, Moradi A, et al. CRAFT: Lightweight Tweakable Block Cipher with Efficient Protection Against DFA Attacks[J]. *IACR Transactions on Symmetric Cryptology*, 2019: 5-45.
- [6] Banik S, Bogdanov A, Isobe T, et al. Midori: A Block Cipher for Low Energy[C]. *International Conference on the Theory and Application of Cryptology and Information Security*. Berlin, Heidelberg: Springer, 2015: 411-436.
- [7] Lai X J, Massey J L, Murphy S. Markov Ciphers and Differential Cryptanalysis[M]. *Advances in Cryptology — EUROCRYPT '91*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007: 17-38.
- [8] Hornauer G, Wernsdorf R, Stephan W. Markov Ciphers and Alternating Groups[C]. *EUROCRYPT '93: Workshop on the theory and application of cryptographic techniques on Advances in cryptology*, 1994: 453-460.
- [9] O'Connor L, Golić J D. A Unified Markov Approach to Differential and Linear Cryptanalysis[M]. *Advances in Cryptology — ASIACRYPT'94*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995: 385-397.
- [10] Sugita M, Kobara K, Uehara K, et al. "Relationships among Differential, Truncated Differential, Impossible Differential Cryptanalyses against Word-Oriented Block Ciphers like RIJNDAEL, E2[C]. *AES Candidate Conference*, 2000: 242-254.
- [11] Ebrahimi Moghaddam A, Ahmadian Z. New Automatic Search Method for Truncated-Differential Characteristics Application to Midori, SKINNY and CRAFT[J]. *The Computer Journal*, 2020, 63(12): 1813-1825.
- [12] Zhang Z Z, Guo X Y, Lin Y. Full Eigenvalues of the Markov Matrix for Scale-Free Polymer Networks[J]. *Physical Review E, Statistical, Nonlinear, and Soft Matter Physics*, 2014, 90(2): 022816.
- [13] Bogdanov A, Knudsen L R, Leander G, et al. PRESENT: An Ultra-Lightweight Block Cipher[C]. *International Workshop on Cryptographic Hardware and Embedded Systems*. Berlin, Heidelberg: Springer, 2007: 450-466.
- [14] Liu B T, Li L, Wu R X, et al. Loong: A Family of Involutional Lightweight Block Cipher Based on SPN Structure[J]. *IEEE Access*, 2019, 7: 136023-136035.
- [15] Knudsen L R. Truncated and Higher Order Differentials[M]. *Fast Software Encryption*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995: 196-211.
- [16] Wu S B, Wang M S. Automatic Search of Truncated Impossible Differentials for Word-Oriented Block Ciphers[C]. *International Conference on Cryptology in India*. Berlin, Heidelberg: Springer, 2012: 283-302.
- [17] Beierle C, Jean J, Kölbl S, et al. The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS[C]. *Annual International Cryptology Conference*. Berlin, Heidelberg: Springer, 2016: 123-153.
- [18] Coppersmith D, Winograd S. Matrix Multiplication via Arithmetic Progressions[J]. *Journal of Symbolic Computation*, 1990, 9(3): 251-280.
- [19] Liang X Y, Huang Z, Wen Q Y. Fast Modular Power Algorithm

Based on Length Limited Run-Length Coding[C]. 2009 Global Mobile Congress, 2009: 1-4.

[20] Cannon L E. A Cellular Computer to Implement the Kalman Filter Algorithm[D]. Bozeman, MT, USA: Montana State University, 1969.



刘慧 于 2014 年在南昌大学计算机科学与技术专业获得学士学位。现在中国科学院大学网络空间安全专业攻读硕士学位。研究领域为密码分析。研究兴趣包括: 分组密码分析、量子计算、量子算法。Email: liuhui0524@iie.ac.cn



杨理 中国科学院信息工程研究所信息安全国家重点实验室研究员、博士生导师, 研究领域为量子计算、量子密码和相关密码学理论。Email: yangli@iie.ac.cn