GRANULE 和 MANTRA 算法的不可能差分区分器分析

武小年[1,2]，李迎新[1,3]，韦永壮[1]，孙亚平[1]

（1. 桂林电子科技大学广西密码学与信息安全重点实验室，广西 桂林 541004；
2. 保密通信重点实验室，四川 成都 610041；3. 广西高校云计算与复杂系统重点实验室，广西 桂林 541004）

摘 要：轻量级分组密码算法 GRANULE 和 MANTRA 结构简单，加密速度快且易于软硬件实现，特别适用于资源受限环境。为对这 2 种算法进行安全性分析，提出一种不可能差分区分器的自动化搜索方法。基于 GRANULE 和 MANTRA 算法结构特性，通过分析其 S 盒的差分分布表得到 S 盒差分特征，再利用中间相遇思想，分别对从加/解密方向得到的差分路径进行遍历，筛选出概率为 0 的最优差分路径。分析结果表明，GRANULE 算法存在 144 个不同的 7 轮不可能差分区分器；MANTRA 算法存在 52 个不同的 9 轮不可能差分区分器。与已有结果相比较，新发现的区分器轮数均是目前最高的。

关键词：轻量级分组密码算法；S 盒；不可能差分区分器；自动搜索

中图分类号：TP309

文献标识码：A

doi:10.11959/j.issn.1000−436x.2020025

Impossible differential distinguisher analysis of GRANULE and MANTRA algorithm

WU Xiaonian[1,2], LI Yingxin[1,3], WEI Yongzhuang[1], SUN Yaping[1]

1. Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, Guilin 541004, China
2. Science and Technology on Communication Security Laboratory, Chengdu 610041, China
3. Guangxi Colleges Key Laboratory of Cloud Computing and Complex Systems, Guilin 541004, China

Abstract: The lightweight block cipher algorithms called GRANULE and MANTRA have a simple structure, fast encryption speed, and they can be easy implemented in software and hardware. Two algorithms are especially suitable for resource-constrained environments. To analyze the security of two algorithms, an automatic search method of impossible differential distinguishers was proposed. Based on the structural characteristics of the GRANALE and MANTRA, the S-box differential characteristics were obtained by analyzing the S-box differential distribution table, and then the idea of intermediate encounter was used to traverse from the difference path obtained from the encryption/decryption direction seperately to select the optimal differential path with probability 0. The analysis results show that there are 144 different 7-round impossible differential distinguishers in the GRANULE, and 52 different 9-round impossible differential distinguishers in the MANTRA. Compared with the existing results, the rounds of the proposed distinguisher is currently the highest.

Key words: lightweight block cipher algorithm, S-box, impossible differential distinguisher, automatic search

---

收稿日期：2019–08–28；**修回日期**：2019–12–12

基金项目：保密通信重点实验室基金资助项目（No.6142103190103）；国家自然科学基金资助项目（No.61572148, No.61872103）；广西科技计划基金资助项目（桂科 No.AB18281019）；广西自然科学基金资助项目（No.2018GXNSFAA294036）；广西密码学与信息安全重点实验室基金资助项目（No.GCIS201705）；广西高校云计算与复杂系统重点实验室基金资助项目（No.YF16205）；广西研究生教育创新计划基金资助项目（No.YCSW2018138, No.YCBZ2018051）

Foundation Items: The Foundation of Science and Technology on Communication Security Laboratory (No.6142103190103), The National Natural Science Foundation of China (No.61572148, No.61872103), The Key Research and Development Plan of Guangxi (guike No.AB18281019), The Natural Science Foundation of Guangxi (No.2018GXNSFAA294036), Guangxi Key Laboratory of Cryptography and Information Security (No.GCIS201705), Guangxi Colleges Key Laboratory of Cloud Computing and Complex Systems (No.YF16205), The Innovation Project of Guangxi Graduate Education (No.YCSW2018138, No.YCBZ2018051)

1　引言

近年来，伴随着电子信息技术的迅速发展，射频识别、传感器网络和智能卡等微型计算设备应用需求不断增大。注意到，这些微型计算设备的存储空间小、计算能力弱、功耗小等资源受限特点使常规的对称密码算法并不适用。如何设计新型的轻量级密码算法，以满足各种资源受限环境下的使用，成为目前的研究热点。过去的 10 年里，多个轻量级算法被相继提出，比如 LBlock、GIFT、Midori、PRESENT、MIBS、LED、SPECK 等密码算法。另一方面，轻量级分组密码的安全性至关重要，常见的密码分析方法包括差分密码分析[1]、线性密码分析[2]、不可能差分分析[3-4]、积分分析[5]等。

不可能差分分析是对差分分析的扩展，由 Knudsen[3]和 Biham 等[4]提出。不可能差分分析的关键在于构造不可能差分区分器，以进行密钥筛选。由于该攻击方法简单、有效，因此广泛应用到分组密码攻击中[6-8]。为了能够快速高效地寻找高轮数的不可能差分区分器，多种自动化搜索方法被先后提出。2012 年，Wu 等[9]提出一个较通用的不可能差分区分器的自动化搜索方法，其将 r 轮分组密码结构视为一个方程组，描述了内部基元中差分的传播行为，尤其是分组密码结构的 S 盒置换或分支交换。2017 年，Luo 等[10]改进 Wu 等[9]提出的自动化搜索方法，并测试该方法是否存在解，其主要简化了最耗时的矩阵运算，在更短的时间内找到了更多不可能的差分，提高了搜索效率。同年，Sasaki 等[11]针对分组密码算法，提出了基于 MILP 模型的比特级的不可能差分自动化搜索方法，并给出了 Midori-128 算法的 7 轮不可能差分区分器。2018 年，韩亚等[12]通过学习基于比特的可分性质，利用三子集传播方程，提出一种基于 SAT/SMT 求解器自动化搜索 ARX 结构分组密码积分区分器的方法。张仕伟等[13]利用 SIMON 中 AND 组件的差分传播特性，构造 2 条约束条件，结合 SAT 求解器提出了自动化搜索算法，搜索出多条 11 轮不可能差分区分器，该算法可以准确地判断 SIMON 算法的任意差分对能否构成一条不可能差分区分器。Zhang 等[14]提出了“Modes operation”方法，用于对 ARX 类型的密码算法进行不可能差分区分器的自动化搜索。

2018 年，Bansod 等提出了 2 种轻量级分组密码算法—— GRANULE[15]和 MANTRA[16]。这 2 种算法均使用了典型的 Feistel 结构，并采用了轻量级 S 盒和简单的位操作，使其能够在较少的轮数中完成最大扩散效果。他们对 GRANULE 和 MANTRA 算法分别进行了安全性分析，表明 2 种算法均能有效地抵抗差分分析、线性分析、Biclique 攻击、零相关分析等。2019 年，石淑英等[17]针对 GRANULE 算法构造了 9 个 5 轮不可能差分区分器，但是该方法在构造不可能差分区分器中并未考虑算法内部核心部件代数性质。如何针对这些新算法构建更高轮数的不可能差分区分器有待深入解决。

本文基于 GRANULE 和 MANTRA 算法结构，利用密码 S 盒差分分布表新性质，结合中间相遇思想，提出了一种不可能差分区分器的新自动化搜索方法。基于该搜索方法，本文发现 GRANULE/MANTRA 算法有 144/52 个不同的 7/9 轮的不可能差分区分器。

2　算法介绍

2.1　GRANULE 算法简介

GRANULE 算法是一种基于 Feistel 结构的轻量级分组密码算法，分组长度为 64 bit，支持 80 bit 和 128 bit 这 2 种密钥长度，迭代轮数为 32 轮。该算法中的轮函数 F 包括置换层、S 盒、循环移位、密钥加和异或运算。GRANULE 算法结构如图 1 所示。

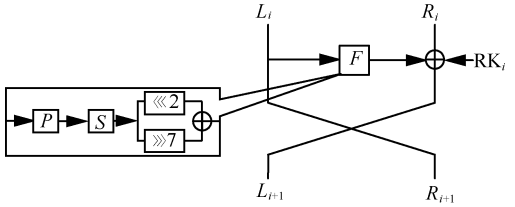

图 1　GRANULE 算法结构

设 GRANULE 算法第 i 轮输入是 C_i，输出是 C_{i+1}，该算法的左分支和右分支分别记为 L_i 和 R_i(i=0,1,2,⋯,31)，则从(L_i,R_i)更新至(L_{i+1},R_{i+1})，更新过程为

$$\begin{cases} C_i = L_i \| R_i \\ L_{i+1} = ((S(P(L_i)) <<< 2) \oplus (S(P(L_i)) >>> 7)) \oplus \mathrm{RK}_i \\ R_{i+1} = L_i \end{cases} \tag{1}$$

GRANULE 算法采用一个 4 bit 的 S 盒，即 $\{0,1\}^4 \rightarrow \{0,1\}^4$。S 盒具体数值以十六进制的形式给出，如表 1 所示。

表 1 GRANULE 算法 S 盒

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

2.2 MANTRA 算法简介

MANTRA 算法是一种基于 Feistel 结构的轻量级分组密码算法，分组长度为 64 bit，支持 80 bit 和 128 bit 这 2 种密钥长度，迭代轮数为 32 轮。该算法中的轮函数 F 是由 2 轮的 Feistel 结构拼接而成，这 2 个 Feistel 结构都包含了 S 盒、密钥加、循环移位和异或运算 4 个基本操作。MANTRA 算法结构如图 2 所示。

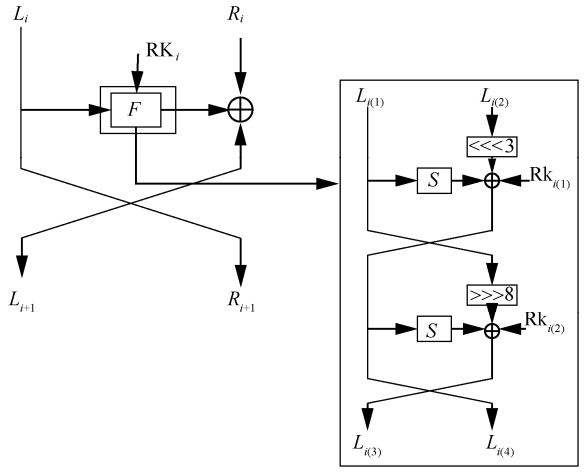

图 2 MANTRA 算法结构

设 MANTRA 算法的第 i 轮输入是 C_i，输出是 C_{i+1}，该算法的左分支和右分支分别记为 L_i 和 $R_i(i=0,1,2,\cdots,31)$，则从(L_i,R_i)更新至(L_{i+1},R_{i+1})，更新过程为

$$\begin{cases} C_i = L_i \| R_i \\ L_i = L_{i(1)} \| L_{i(2)} \\ L_{i(3)} = ((S(S(L_{i(1)}) \oplus (L_{i(2)} <<< 3) \oplus \mathrm{Rk}_{i(1)}) \oplus \\ (L_{i(1)} >>> 8) \oplus \mathrm{Rk}_{i(2)}) \\ L_{i(4)} = (S(L_{i(1)}) \oplus (L_{i(2)} <<< 3) \oplus \mathrm{Rk}_{i(1)}) \\ L_{i+1} = L_{i(3)} \| L_{i(4)} \oplus R_i \\ R_{i+1} = L_i \end{cases} \tag{2}$$

MANTRA 算法中使用的是一个 4 bit 的 S 盒，即 $\{0,1\}^4 \rightarrow \{0,1\}^4$。S 盒具体数值以十六进制的形式给出，如表 2 所示。

表 2 MANTRA 算法 S 盒

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

3 2 种算法的不可能差分区分器构造

2014 年，Tezcan[18]提出一种针对 S 盒评估以及 S 盒差分传播的新标准。当给定 S 盒的差分输入值时，对应 S 盒的输出差分中至少有 1 bit 的概率为 1，即可以确定该输出差分中的 1 bit 的差分值。被确定概率为 1 的比特被称为未受干扰比特。基于该思想，对密码算法中 S 盒的差分分布表进行分析，通过分析输入/输出差分值，找到未受干扰比特。将该方法

应用于不可能差分分析中，可获得更长的不可能差分区分器。

针对 GRANULE 和 MANTRA 算法的不可能差分区分器自动搜索方法，主要是通过分析 2 种算法 S 盒的差分分布表，得到 2 种算法对应的 S 盒差分特征，再利用中间相遇思想，对 2 种算法分别从加/解密方向得到的差分路径进行遍历，筛选出概率为 0 的最优差分路径，即不可能差分区分器。

3.1　S 盒的差分特征

Biham 等[1]对 DES 算法进行差分密码分析的关键在于观察到 S 盒差分分布不均匀特性，并给出了 S 盒差分分布表的概念，这个概念完全刻画了 S 盒的差分传播特征，实际上也是满足特定差分的随机输入对经过 S 盒作用后输出差分的分布特征。基于文献[18]中使用的方法，对差分分布表输入/输出差分进行分析总结，得到 S 盒的输入/输出差分特征，将该特征应用到不可能差分区分器的推导过程中，可有效提高不可能差分区分器的长度。

定义 1　S 盒差分分布表。设 $i,j\in N$，给定 $m\in F_2^i$，$n\in F_2^j$，从 F_2^i 到 F_2^j 的非线性映射（也称为 S 盒）定义为

$$\begin{cases} IN_S(m,n)=\{x\in F_2^i : S(x\oplus m)\oplus S(x)=n\} \\ N_S(m,n)=\#IN_S(m,n) \end{cases} \tag{3}$$

构造 $2^i\times 2^j$ 的表格如下：以 m 为行指标遍历 F_2^i，n 为列指标遍历 F_2^j，行列交错处的取值 $N_S(m,n)$。称 m 为 S 盒的输入差分，n 为 S 盒的输出差分。三元数组$(m,n,N_S(m,n))$按上述方式构成的表即为 S 盒的差分分布表。

3.1.1　GRANULE 算法 S 盒的差分特征

根据定义 1，构造 GRANULE 算法 S 盒的差分分布表，GRANULE 算法 S 盒的差分分布表如表 3 所示。

当 S 盒的输入/输出差分为某些定值时，其输入/输出差分会存在一定的规律性。如输入差分为 0001 时，输出差分可能为 1000，1001，1010，1011，1100，1101，1110，1111。而这些输出差分的第 3 比特均为 1，另外 3 个比特的差分未知，简记为 1***（其中“*”表示未知差分）。根据该方法，对表 3 进行总结可得到性质 1。

性质 1　GRANULE 算法 S 盒的差分分布表输入/输出差分不均匀的特征表明，当 S 盒的输入差分为某些定值时，其输出差分存在相应的特征，其输出差分存在的传播特性如表 4 所示。

表 3　GRANULE 算法 S 盒的差分分布表

输入差分	输出差分															
	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	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2
2	0	0	0	4	0	0	4	0	0	0	2	2	0	0	2	2
3	0	4	0	0	0	4	0	0	0	0	2	2	0	0	2	2
4	0	0	0	4	0	0	4	0	0	0	2	2	0	0	2	2
5	0	0	0	0	0	0	4	4	2	2	0	0	2	2	0	0
6	0	0	0	0	0	0	0	0	4	4	0	0	4	4	0	0
7	0	4	0	0	0	4	4	4	0	0	0	0	0	0	0	0
8	0	0	0	2	2	2	0	2	0	0	0	2	2	2	0	2
9	0	2	4	0	2	0	0	0	2	0	0	0	0	2	0	4
a	0	0	0	2	2	2	0	2	0	0	0	2	2	0	2	0
b	0	2	4	0	2	0	0	0	0	2	0	0	2	0	4	0
c	0	0	0	2	2	2	0	2	2	2	0	2	0	0	0	2
d	0	2	4	0	2	0	0	0	0	2	0	4	2	0	0	0
e	0	0	0	2	2	2	0	2	2	2	2	0	0	0	2	0
f	0	2	4	0	2	0	0	0	2	0	4	0	0	2	0	0

表 4　GRANULE 算法 S 盒输入/输出差分特征

输入差分	输出差分
0001	1***
0010	**1*
0100	**1*
0110	1*0*
0111	0***

表 4 中，“*”表示差分未知，“0”表示差分为 0，“1”表示差分为 1。将表 4 中 GRANULE 算法 S 盒的差分特征应用于该算法的加/解密过程中，可有效拓展不可能差分区分器的长度。

3.1.2　MANTRA 算法 S 盒的差分特征

根据定义 1，构造 MANTRA 算法 S 盒的差分分布表，然后利用 3.1.1 节中的方法对 S 盒的差分分布表进行分析可得到性质 2。

性质 2　MANTRA 算法 S 盒的差分分布表输入/输出差分不均匀的特征表明，当 S 盒的输入差分为某些定值时，其输出差分存在相应的特征，其输出差分存在的传播特性如表 5 所示。

表 5　MANTRA 算法 S 盒的输入/输出差分特征

输入差分	输出差分
0010	1***
1101	1***
1111	0***

将表 5 中 MANTRA 算法 S 盒的差分特征应用于该算法的加/解密过程中，可有效拓展不可能差分区分器的长度。

3.2 GRANULE 算法不可能差分区分器自动搜索方法

从 GRANULE 算法 S 盒差分分布表的输入/输出差分中找到 S 盒的输入/输出差分传播特征，在此基础上，基于中间相遇思想，先从加密方向找到一条概率为 1 的有效差分路径，然后从解密方向找到一条概率为 1 的有效差分路径；再在上述的加密方向的路径集合和解密方向的路径集合中进行遍历，并将其拼接，筛选出一条概率为 0 的最优差分路径，即不可能差分区分器。在搜索筛选概率为 0 的最优差分路径过程中，采用自动化搜索的方式进行遍历。

中间相遇思想是构造不可能差分区分器的常用方法，将一个密码算法 T 分成两部分：$T=T_0T_1$，T_0 存在概率为 1 的差分 $\Delta E \to \Delta F$，T_1 存在概率为 1 的差分 $\Delta G \to \Delta H$；如果 ΔF 和 ΔH 不相等，那么对于 T 就存在概率为 0 的差分 $\Delta E \to \Delta G$，即 $\Delta E \to \Delta G$ 称为“不可能差分区分器”，如图 3 所示。

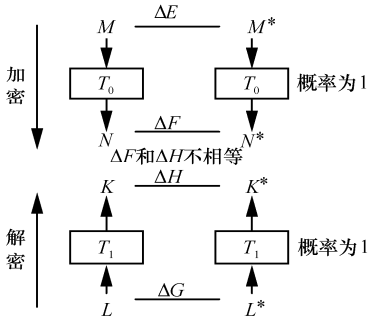

图 3 不可能差分区分器原理

3.2.1 GRANULE 算法有效差分路径

对 GRANULE 算法有效差分路径的搜索，首先是在算法的加密方向，利用中间相遇思想找到一条概率为 1 的差分路径，具体地，当输入差分在经过轮函数中 S 盒时，利用性质 1 中 S 盒的差分特征进行差分传播，输出较为准确的差分，直到输出差分全为未知差分时，则停止搜索加密方向的差分路径，并输出差分路径集合。其次，在算法的解密方向，按照同样的方式，从相反的解密方向进行搜索，最终输出解密方向的差分路径集合。

GRANULE 算法加密方向搜索概率为 1 的差分路径过程如算法 1 所示。

算法 1 GRANULE 算法加密方向有效差分路径自动搜索

输入 64 bit 差分明文 M，其左右分支分别记为 L_i 和 $R_i(i=0,1,2,\cdots,31)$

输出 输出每轮经过轮函数的输出差分集合 List

1) 初始化；
2) $L||R \leftarrow M$；
3) when M is not all “*”
4) { List.append (M)；
5) $p \leftarrow P(L)$；
6) $s \leftarrow p$； //将 p 拆分成 8 个半字节
7) for (i=0,i<= 7,i++) //利用性质 1 进行差分传播
8) { for (j=0,j<= 5,j++)
9) { if $s[i]$ is input[j] // input 和 output 分别为性质 1 中 S 盒的输入/输出差分特征
10) $s[i] \leftarrow$output[j]；
11) else if $s[i]$ is not“0000”
12) $s[i] \leftarrow$“****”；
13) }
14) }
15) $m \leftarrow s$； //将 s 中 8 个半字节合并在一起
16) shift←Shift(m)； //循环移位
17) $R \leftarrow L$；
18) $L \leftarrow$shift；
19) $M \leftarrow L||R$；
20) }
21) return List；

3.2.2 GRANULE 算法不可能差分区分器

基于上述 GRANULE 算法加/解密方向自动搜索的概率为 1 的差分路径集合，利用中间相遇思想，对算法 1 得到的加密方向的差分路径集合和解密方向的差分路径集合进行遍历拼接，搜索一条概率为 0 的最优差分路径，即不可能差分区分器。

GRANULE 算法搜索不可能差分区分器过程如算法 2 所示。

算法 2 GRANULE 算法不可能差分区分器自动搜索

输入 加密方向差分路径集合 List_en，解密方向差分路径集合 List_de

输出 输出最优不可能差分区分器轮数

1) 初始化；
2) int result = 0；

3) for (*i*=0,*i*<List_en.size(),*i*++)　//加密方向差分路径集合

4) { for (*j*= List_de.size()–1,*j*>=0, *j*– –)　//解密方向差分路径集合

5)　{ if contradiction(List_en.get(*i*), List_de.get(*j*))　//List_en.get(*i*)为获取加密方向的一条输入路径，List_de.get(*j*)为获取解密方向的一条输入路径，contradiction 为检测 2 个输入是否存在矛盾点，存在返回 true，否则返回 false

6)　　result = result>(*i*+*j*)?result:(*i*+*j*);

7)　}

8) }

9) return result;

算法 2 通过遍历算法 1 中得到的加密方向差分路径集合和解密方向差分路径集合，每次分别从加密方向和解密方向路径中选取一条数据，利用 contradiction 函数进行矛盾点的检测。通过遍历和检测，最终输出最优不可能差分区分器轮数。

由于 GRANULE 算法的分组长度为 64 bit，遍历所有可能差分的复杂度太高，因此本文只对具有 1 bit 活跃的输入/输出差分对进行搜索。利用上述自动搜索方法进行搜索，搜索到了 144 个不同的 7 轮不可能差分区分器，图 4 给出了其中一条不可能差分区分器的具体形式。

表 6 列出了所搜索出的 GRANULE 算法部分 7 轮不可能差分区分器，其中，v_i(0≤*i*≤7)表示第 *i* bit 的差

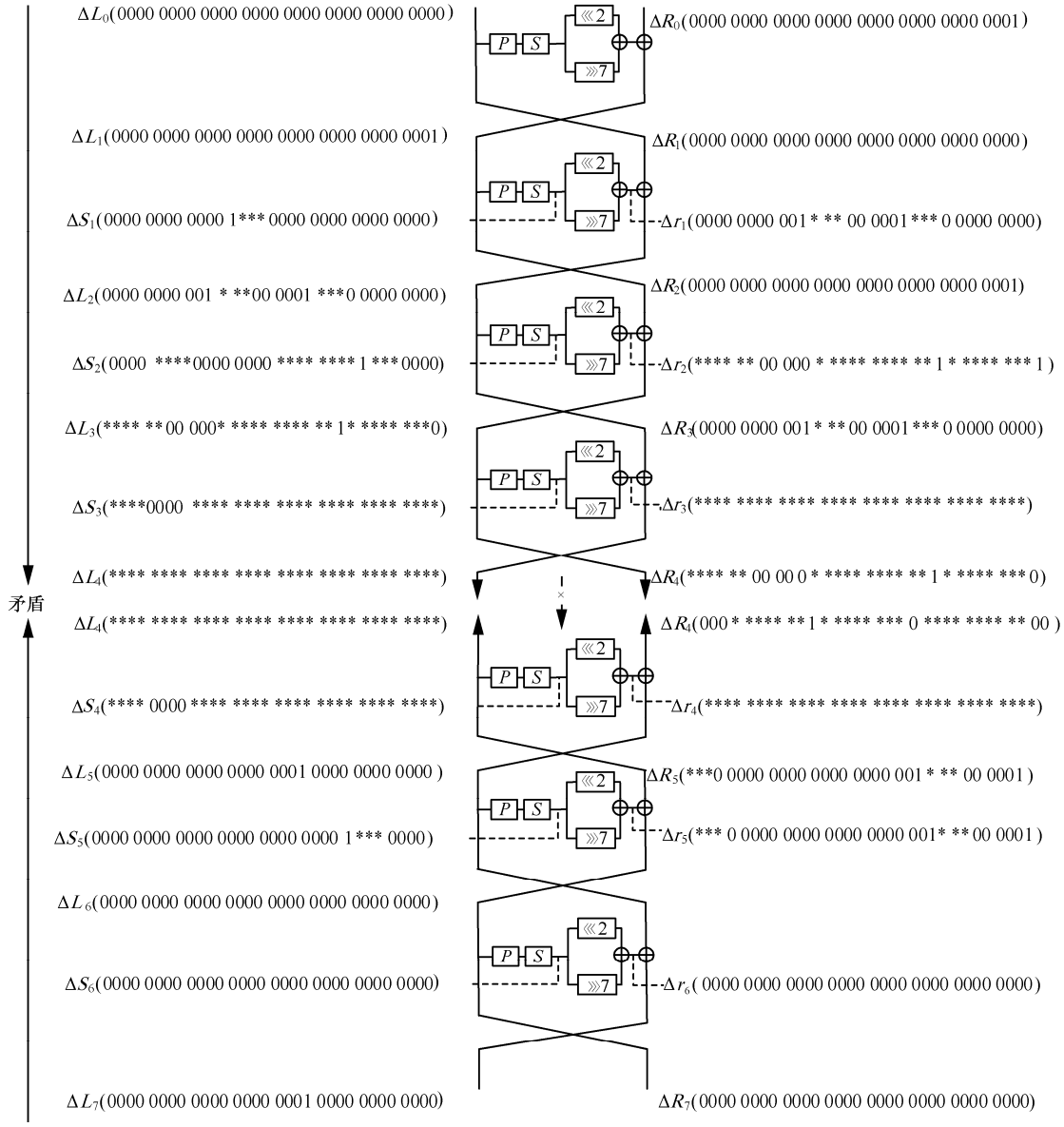

图 4　GRANULE 算法不可能差分区分器示意

分为 1，其余比特差分为 0，0 表示该字节的差分为 0。

表 6 GRANULE 算法不可能差分区分器

不可能差分输入($\Delta L_0 \| \Delta R_0$)	不可能差分输出($\Delta L_7 \| \Delta R_7$)
$0,0,0,0,v_6,0,0,0$	$0,0,0,v_4,0,0,0,0$
$0,0,0,0,v_5,0,0,0$	$0,0,0,v_4,0,0,0,0$
$0,0,0,0,0,0,0,v_0$	$0,v_0,0,0,0,0,0,0$
$0,0,0,0,0,0,0,v_0$	$0,0,v_4,0,0,0,0,0$
$0,0,0,0,v_7,0,0,0$	$0,0,0,v_4,0,0,0,0$

3.3 MANTRA 算法不可能差分区分器自动搜索方法

MANTRA 算法不可能差分区分器的搜索方法与 GRANULE 算法不可能差分区分器的搜索方法类似，是根据 MANTRA 算法的差分特征，再采用中间相遇思想，分别从加/解密方向各找到一条概率为 1 的有效差分路径；再对加/解密方向的路径集合进行遍历和拼接，筛选出一条概率为 0 的最优差分路径，即不可能差分区分器。2 种算法搜索方法不同的地方在于 2 种算法 S 盒的差分特征不同，因此 MANTRA 算法在搜索加/解密方向的有效差分路径时，需要根据其自身算法 S 盒的差分特征进行传播，从而输出对应的差分路径。

由 3.2 节中提出的不可能差分区分器自动化搜索方法，利用性质 2 中的 MANTRA 算法 S 盒的输入/输出差分特征，只对具有 1 bit 活跃的输入/输出差分对进行自动化搜索，可搜索到 52 个不同的 9 轮不可能差分区分器，表 7 列出了所搜索出的 MANTRA 算法部分 9 轮不可能差分区分器，其中，$v_i(0 \leqslant i \leqslant 7)$表示第 i bit 的差分为 1，其余比特差分为 0，0 表示该字节的差分为 0。

表 7 MANTRA 算法不可能差分区分器

不可能差分输入($\Delta L_0 \| \Delta R_0$)	不可能差分输出($\Delta L_7 \| \Delta R_7$)
$0,0,0,0,0,0,0,v_3$	$0,0,0,v_1,0,0,0,0$
$0,0,0,0,0,0,0,v_2$	$0,v_5,0,0,0,0,0,0$
$0,0,0,0,0,0,0,v_1$	$0,v_5,0,0,0,0,0,0$
$0,0,0,0,0,0,0,v_1$	$v_1,0,0,0,0,0,0,0$
$0,0,0,0,0,0,0,v_1$	$0,v_4,0,0,0,0,0,0$

4 分析结果对比

为加强对 GRANULE 算法和 MANTRA 算法的安全性分析，并给出本文不可能差分区分器对这 2 种算法的分析结果，采用 Java 语言实现了本文提出的搜索方法，并在处理器为 Intel i5-8500，内存为 8 GB 的 Windows10 家庭版系统环境下运行，算法的时间复杂度为 $O(n^4)$，其中，n 表示输入的明文长度。针对 GRANULE 算法和 MANTRA 算法，遍历 1 bit 活跃的输入输出差分对，测试所使用的时间分别为 392 ms 和 274 ms。

为进一步归纳对这 2 种算法的现有方法安全性分析结果，将本文分析结果与现有文献进行了对比。针对 GRANULE 算法的安全性分析，将本文结果与文献[10,17]中提出的自动化搜索方法，以及文献[15]采用的零相关线性分析进行对比。零相关线性分析由 Bogdanov 等[19]提出，该分析方法是寻找密码算法概率为 $\frac{1}{2}$，即相关性为 0 的线性逼近作为零相关线性分析的区分器，进而区分出正确密钥和错误密钥。零相关线性分析作为不可能差分分析的对偶方法，存在与不可能差分区分器对比的合理性。针对 MANTRA 算法的安全性分析，将本文结果与文献[10]中提出的自动化搜索方法，以及文献[16]中的零相关线性分析进行了对比。对 GRANULE 算法和 MANTRA 算法的分析结果如表 8 所示。

表 8 对 GRANULE 算法和 MANTRA 算法的分析结果

算法名称	分析方法	区分器轮数	个数	对比文献
GRANULE	不可能差分分析	5	9	文献[17]
	不可能差分分析	7	144	本文方法
	不可能差分分析	6	38	文献[10]
	零相关线性分析	6	—	文献[15]
MANTRA	不可能差分分析	9	52	本文方法
	不可能差分分析	6	52	文献[10]
	零相关线性分析	8	—	文献[16]

文献[15]在提出 GRANULE 算法时，使用零相关线性分析的分析方法对其进行安全性分析，构造出 6 轮的零相关线性区分器。文献[17]针对 GRANULE 算法给出了 9 条 5 轮不可能差分区分器中，其 S 盒的输入/输出差分仅考虑“0”与“非 0”这 2 种情况，即输入差分为“0”时，输出差分也为“0”；输入差分为“非 0”时，输出差分为“****”。文献[10]对 GRANULE 算法的自动化搜索，得到 38 个 6 轮不可能差分区分器，其采用线性运算，以半字节进行搜索，搜索时间为 76 ms。本文通过对 GRANULE 算法中 S 盒的差分分布表的输入/输出差分进行总结，得到性质 1 所示的 S 盒差分特征，

其能够获得输入/输出差分的传播规律，如在非 0 情况下，若输入差分为“0001”其输出差分为“1***”；将该性质与中间相遇思想结合，采用自动化搜索方法，以比特进行搜索，搜索时间较文献[10]更长，但获得的不可能差分区分器的轮数也更长。

文献[16]在提出 MANTRA 算法时，使用零相关线性分析的分析方法对其进行安全性分析，构造出 8 轮的零相关线性区分器。文献[10]对 MANTRA 算法的自动化搜索，得到 52 个 6 轮不可能差分区分器，其采用线性运算，以半字节进行搜索，其搜索时间为 51 ms。本文通过对 MANTRA 算法 S 盒的差分分布表的输入/输出差分进行总结，得到性质 2 所示的 S 盒差分特征，通过结合中间相遇思想，并采用自动化搜索方法，以比特进行搜索，搜索时间较文献[10]更长，但获得的不可能差分区分器的轮数也更长。

综上所述，通过分析密码算法 S 盒的差分分布表的输入/输出差分特征，并采用自动化搜索方法，结合中间相遇思想，可以提高算法的不可能差分区分器长度。

5　结束语

本文针对GRANULE和MANTRA算法的结构特性，通过分析其 S 盒差分分布表获得对应的差分特征性质，结合中间相遇思想，采用自动化搜索的方式，在加/解密方向分别找到一条概率为 1 的有效差分路径；再在上述的加/解密方向的路径集合中进行遍历，找到一条概率为 0 的最优差分路径。研究结果发现，GRANULE/MANTRA 算法有 144/52 个不同的 7/9 轮的不可能差分区分器。在后续工作中，将充分地考虑密码核心部件的代数性质，以获得更高轮数的区分器。

参考文献：

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

[2] MATSUI M. Linear cryptanalysis method for DES cipher[C]// Workshop on the Theory and Application of of Cryptographic Techniques. 1993: 386-397.

[3] KNUDSEN L. DEAL-a 128-bit block cipher[J]. Complexity, 1998, 258(2): 216.

[4] BIHAM E, BIRYUKOV A, SHAMIR A. Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials[C]//EUROCRYPT'99. 1999: 12-23.

[5] KNUDSEN L, WAGNER D. Integral cryptanalysis[C]//International Workshop on Fast Software Encryption. 2002: 112-127.

[6] LI M M, GUO J S, CUI J Y, et al. Impossible differential cryptanalysis of speck[C]//Chinese Conference on Trusted Computing and Information Security. 2018: 16-31.

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

[8] 陈平, 廖福成, 卫宏儒. 对轻量级密码算法 MIBS 的相关密钥不可能差分攻击[J]. 通信学报, 2014, 35(2): 190-193+201.
CHEN P, LIAO F C, WEI H R. Related-key impossible differential attack on a lightweight block cipher MIBS[J]. Journal on Communications, 2014, 35(2): 190-193+201.

[9] WU S B, WANG M S. Automatic search of truncated impossible differentials for word-oriented block ciphers[C]//International Conference on Cryptology. 2012: 283-302.

[10] LUO Y Y, LAI X J. Improvements for finding impossible differentials of block cipher structures[J]. Security and Communication Networks, 2017, 2017: 1-9.

[11] SASAKI Y, TODO Y. New impossible differential search tool from design and cryptanalysis aspects[C]//Annual International Conference on the Theory and Applications of Cryptographic Techniques. 2017: 185-215.

[12] 韩亚, 王明生. ARX 结构分组密码积分区分器的自动化搜索[J]. 通信学报, 2018, 39(5): 103-110.
HAN Y, WANG M S. Automatic method for searching integral distinguishers of ARX block ciphers[J]. Journal on Communications, 2018, 39(5): 103-110.

[13] 张仕伟, 陈少真. SIMON 不可能差分及零相关路径自动化搜索算法[J]. 软件学报, 2018, 29(11): 3544-3553.
ZHANG S W, CHEN S Z. Automatic search algorithm for impossible differential trials and zero-correlation linear trials in SIMON[J]. Journal of Software, 2018, 29(11): 3544-3553.

[14] ZHANG K, GUAN J, HU B. Automatic search of impossible differentials and zero-correlation linear hulls for ARX ciphers[J]. China Communications, 2018, 15(2): 54-66.

[15] BANSOD G, PATIL A, PISHAROTY N. GRANULE: an ultra lightweight cipher design for embedded security[R]. Cryptology ePrint Archive, Report 2018/600, 2018.

[16] BANSOD G, PISHAROTY N, PATIL A. MANTRA: an ultra lightweight cipher design for ubiquitous computing[J]. International Journal of Ad Hoc and Ubiquitous Computing, 2018, 28(1): 13-26.

[17] 石淑英, 何骏. GRANULE 算法的不可能差分分析[J]. 计算机工程, 2019, 45(10): 134-138.
SHI S Y, HE J. Impossible differential cryptanalysis of GRANULE[J]. Computer Engineering, 2019, 45(10): 134-138.

[18] TEZCAN C. Improbable differential attacks on present using undisturbed bits[J]. Journal of Computational & Applied Mathematics, 2014, 259(259): 503-511.

[19] BOGDANOV A, RIJMEN V. Linear hulls with correlation zero and linear cryptanalysis of block ciphers[J]. Designs, Codes and Cryptography, 2014, 70(3): 369-383.

[作者简介]

武小年（1972– ），男，湖北监利人，桂林电子科技大学副教授，主要研究方向为分布式计算、信息安全。

李迎新（1991– ），男，河南南阳人，桂林电子科技大学硕士生，主要研究方向为信息安全。

韦永壮（1976– ），男，壮族，广西百色人，博士，桂林电子科技大学教授，主要研究方向为密码学、信息安全。

孙亚平（1993– ），女，山东菏泽人，桂林电子科技大学硕士生，主要研究方向为信息安全。