Modeling and Simulation 建模与仿真, 2024, 13(3), 2568-2578
Published Online May 2024 in Hans. https://www.hanspub.org/journal/mos
https://doi.org/10.12677/mos.2024.133234

Hans汉斯

基于MILP的BORON中间相遇分析

付　豪[1]，刘　亚[1]，赵逢禹[2]，曲　博[3]

[1]上海理工大学光电信息与计算机工程学院，上海
[2]上海出版印刷高等专科学校信息与智能工程系，上海
[3]广东科技学院计算机学院，广东　东莞

收稿日期：2024年4月18日；录用日期：2024年5月17日；发布日期：2024年5月23日

摘　要

BORON是新设计的超轻量级分组密码，被广泛应用于资源受限的设备中保护数据安全，然而受计算和存储资源限制，设计者可能为了追求高的软硬件实现效率而适度降低安全性。为了保障它在实际系统中有足够安全强度，需要评估BORON抵抗各种密码分析方法的能力。本文分析了BORON抵抗中间相遇攻击的能力。具体来说，利用混合整数线性规划自动化搜索算法，找到多条5轮中间相遇差分链，在此基础上构造了9轮的中间相遇分析路径，基于该路径最后恢复9轮BORON的密钥。整个攻击需要的时间、数据和存储复杂度分别为$2^{95.84}$次9轮加密、$2^{42.00}$个选择明文和$2^{94.90}$个64比特块。此结果是对BORON安全性分析的重要补充。

关键词

轻量级分组密码，中间相遇攻击，SPN结构，混合整数线性规划，BORON

MILP-Based for Middle-Meeting Cryptanalysis of BORON

Hao Fu[1], Ya Liu[1], Fengyu Zhao[2], Bo Qu[3]

[1]Department of Computer Science and Engineering, University of Shanghai for Science and Technology, Shanghai
[2]Department of Information and Intelligence Engineering, Shanghai Publishing and Printing College, Shanghai
[3]School of Computer Science, Guangdong University of Science and Technology, Dongguan Guangdong

Received: Apr. 18th, 2024; accepted: May. 17th, 2024; published: May. 23rd, 2024

Abstract

BORON is a recently designed ultra-lightweight block cipher that is widely used to protect data

文章引用: 付豪, 刘亚, 赵逢禹, 曲博. 基于 MILP 的 BORON 中间相遇分析[J]. 建模与仿真, 2024, 13(3): 2568-2578.
DOI: 10.12677/mos.2024.133234

security in resource-constrained devices; however, due to computational and storage resource constraints, designers may moderately reduce the security in order to pursue high hardware and software implementation efficiency. In order to guarantee its sufficient security strength in real systems, it is necessary to evaluate the ability of BORON to resist various cryptanalysis methods. In this paper, we analyse the ability of BORON to resist the middle-meeting attacks. Specifically, multiple 5-round middle-meeting differential chains are found using mixed-integer linear programming automated search algorithms, the 5 rounds of middle-meeting differential distinguisher based on the 9 rounds of middle-meeting differential attack, the 9 rounds of middle-meeting differential cryptanalysis of the BORON-128 requires time, data, memory complexities of $2^{95.84}$ 9-round encryption times, $2^{42.00}$ chosen plaintext and $2^{94.90}$ 64-bit blocks; The attack results providing an important supplement to the security analysis of BORON.

Keywords

Lightweight Block Cipher, The Middle-Meeting Attack, The SPN Structure, Mixed Integer Linear Programming, BORON

1. 引言

随着物联网技术的快速发展，物联网设备已经广泛应用于各个领域，这些设备存储了大量的敏感的数据信息，因此，保证这些数据的安全成为了重中之重。现如今，加密技术成为确保数据安全的关键手段之一。但传统加密算法在计算资源有限的物联网环境中显得笨重且不够高效。为了解决这一问题，轻量级分组密码应运而生。

轻量级分组密码是一种专门设计用于计算资源受限环境的加密算法。它们具有计算量小、存储空间小和功耗低的特点，非常适合嵌入式系统、智能卡和物联网设备等场景。在轻量级分组密码的发展历程中，诸如 Piccolo [1]、PRESENT [2]、Prince [3]、Lblock [4]、CRAFT [5]等算法均有着广泛的应用场景。但近年来信息安全事件不断涌现，使得数据安全保障工作越发重要。轻量级分组密码在设计时往往为了追求高的软硬件实现效率，常常会牺牲部分安全性，这可能成为数据安全保障的一个薄弱环节，因此研究轻量级分组密码的安全性非常重要。

BORON[6]算法是 2017 年由 Gaurav BANSOD 等人提出的超轻量级 SPN (Substitution-Permutation Network)结构的分组密码，共 25 轮，最后一轮输出结果与白化密钥进行异或得到密文。BORON 有两个版本，分别为主密钥长度为 80 比特的 BORON-80 和主密钥长度为 128 比特的 BORON-128。目前已经有一些针对 BORON 的安全性研究。Liang [7]等在 2019 年使用自动化搜索算法搜寻到了 8 轮 BORON-80 的差分区分器和 9 轮的 BORON-128 的线性区分器，并在此基础上提出了对 9 轮 BORON-80 差分攻击和 11 轮的 BORON-128 的线性攻击。攻击 9 轮的 BORON-80 需要 256 次 9 轮 BORON-80 加密、263 个选择明文和 224 个 64 比特块；攻击 11 轮的 BORON-128 需要 2123 次 11 轮 BORON-128 加密、263 个选择明文和 242 个 64 比特块。Li [8]等在 2020 年使用基于 MILP (混合整数线性规划)的自动化搜索算法搜寻到了 6 轮的积分区分器，并在此基础上提出了对 7 轮、8 轮和 9 轮 BORON 的积分攻击。攻击 7 轮 BORON 需要 254.19 次 7 轮 BORON 加密、254 个选择明文，攻击 8 轮 BORON 需要 258.34 次 8 轮 BORON 加密、256.32 个选择明文，攻击 9 轮 BORON 需要 294.06 次 9 轮 BORON 加密、257.90 个选择明文，但文中并

未指出需要多少存储复杂度。Wu [9]等在 2021 年尝试对 BORON 进行不可能差分攻击，但文中出现了非零输入经过 S 盒后会得到零输出的错误，会对攻击过程及结果造成影响，因此本文不考虑这篇文章的结果。目前，对于 BORON-128 仅有差分、线性分析和积分攻击。

近年来，中间相遇分析[10]已经成功地评估了众多知名分组密码的安全性，如美国国家标准与技术研究院(NIST)提出的加密标准算法 AES [11]等，因此中间相遇攻击已成为非常重要的密码分析方法。同时，一些自动化搜索模型如 MILP [12]可以进一步提高中间相遇攻击的效率。但是截止目前还没有公开发表的 BORON 抵抗中间相遇攻击的能力的研究。

本文通过使用 MILP 求解模型寻找到了数条 5 轮 BORON 的中间相遇差分链，从中选取预计算过程中需要猜测参数个数最少的中间相遇区分链。在此区分链前端扩展 1 轮，后端扩展 3 轮形成 9 轮的 BORON 中间相遇攻击路径。对 9 轮 BORON-128 的中间相遇攻击所需的数据复杂度为 2^{42} 个选择明文，时间复杂度为 $2^{95.84}$ 次 9 轮 BORON 加密，存储复杂度为 $2^{94.90}$ 个 64 比特块。本文评估了 BORON 抵抗中间相遇攻击的能力，是 BORON 的安全性研究的重要补充。

2. 预备知识

2.1. BORON 算法介绍

BORON 算法是 2017 年由 Gaurav BANSOD 等人提出的超轻量级 SPN 结构的分组密码，共 25 轮，最后一轮输出结果与白化密钥进行异或得到密文。BORON 有两个版本，分别为主密钥长度为 80 比特的 BORON-80 和主密钥长度为 128 比特的 BORON-128。

轮函数 将 64 比特的输入划分为 4 个相同大小的块，每个块 16 比特，以每个块为单位进行轮函数迭代。BORON 算法的轮函数如图 1 所示，轮函数由轮密钥、置换层和线性层组成，每一轮的输入先和轮密钥进行异或，得到的结果进入 16 个相同的 4 比特的 S 盒进行非线性操作，S 盒如表 1 所示，S 盒的输出结果依次进行分组置换、移位操作、异或操作，最终产生密文。

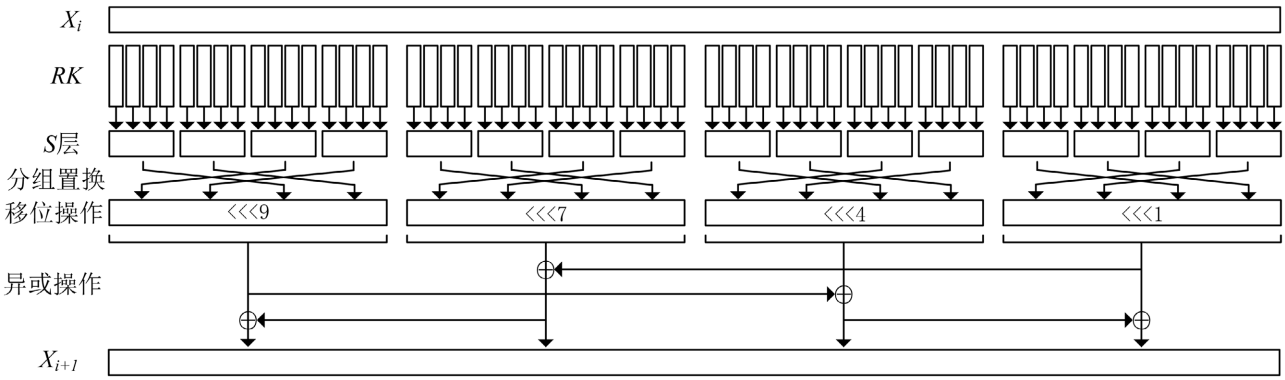

Figure 1. The round function of BORON
图 1. BORON 轮函数

Table 1. 4-bits S box
表 1. 4-bits S 盒

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

密钥调度 将 BORON-128 的 128 比特的主密钥 K 记为 $k_{127}\|k_{126}\|\cdots\|k_1\|k_0$，并把主密钥 K 的低位

64 比特作为初始轮密钥 RK_0，然后将初始轮密钥经由 S 盒和移位操作，产生每一轮新的密钥，最终扩展成 25 轮的轮密钥 RK，具体的密钥扩展算法如算法 1 所示。

算法 1：BORON-128 密钥调度算法

输入：主密钥 K

输出：轮密钥 RK

$k_{127}||k_{126}||\ldots||k_1||k_0 \leftarrow K$ /*将 128 位主密钥 K 记为 $k_{127}||k_{126}||\ldots||k_1||k_0$*/

$RK_0 \leftarrow k_{63}||k_{62}||\ldots||k_1||k_0$ /*取低位 64 比特作为初始轮的轮密钥 RK_0*/

for i1 to 24/*i 从 1 遍历到 24，生成 24 轮轮密钥*/

$k_3||k_2||k_1||k_0 \leftarrow S(k_3||k_2||k_1||k_0)$/*$k_3||k_2||k_1||k_0$ 输入 S 盒，将 S 盒输出作为新的 $k_3||k_2||k_1||k_0$*/

$k_7||k_6||k_5||k_4 \leftarrow S(k_7||k_6||k_5||k_4)$/*$k_7||k_6||k_5||k_4$ 输入 S 盒，将 S 盒输出作为新的 $k_7||k_6||k_5||k_4$*/

$k_{63}||k_{62}||k_{61}||k_{60}||k_{59} \leftarrow k_{63}||k_{62}||k_{61}||k_{60}||k_{59} \oplus i$/*$k_{63}||k_{62}||k_{61}||k_{60}||k_{59}$ 和轮数 i 异或生成新的 $k_{63}||k_{62}||k_{61}||k_{60}||k_{59}$*/

$k_{127}||k_{126}||\ldots||k_1||k_0 \leftarrow k_{127}||k_{126}||\ldots||k_1||k_0 <<<13$ /*将得到的密钥循环左移 13 位*/

$RK_i \leftarrow k_{63}||k_{62}||\ldots||k_1||k_0$ /*取低位 64 比特作为每轮的轮密钥*/

$RK \leftarrow RK_0||RK_1||\ldots||RK_{24}$ /*生成 25 轮的密钥*/

2.2. 符号标记

1) M、C、K：明文、密文和主密钥；

2) M^i：第 i 个明文；

3) $X_i[j]$：第 i 轮输出的第 j 个比特，$1 \le i \le 25$，$0 \le j \le 63$，最右边为最低位；

4) $RK_i[j]$：第 i 轮使用的子密钥的第 j 个比特，$1 \le i \le 25$，$0 \le j \le 63$，最右边为最低位；

5) $Y_i[j], \Delta Y_i[j]$：第 i 轮轮密钥加密输出的第 j 个比特位置的值和差分；

6) $Z_i[j], \Delta Z_i[j]$：第 i 轮 S 盒输出的第 j 个比特位置的值和差分；

7) $W_i[j], \Delta W_i[j]$：第 i 轮分组置换、移位操作输出的第 j 个比特位置的值和差分；

8) $Y_i^r[j]$：第 i 轮 Y^r 在第 j 个比特位置的值；

9) Ω：S 盒输入和输出差分模式的向量；

10) *：此半字节处的差分是未知的；

11) $\oplus$：异或操作；

12) $<<<i$：循环左移 i 比特。

2.3. 中间相遇分析简介

Hellman 和 Diffie 在 1977 年首次提出了中间相遇攻击方法，而后中间相遇攻击方法一直发展至今，并和许多新技术结合形成了更有效的攻击方法。中间相遇攻击中有一种近年来非常有效的攻击思想是将加密算法分成三个部分：区分器、区分器前端和区分器后端。整个攻击过程分为离线部分和在线部分。离线部分主要是在区分器输入端部分选取一个活动的位置构建 δ-集，再将其进行几轮加密后，计算区分器输出端的有序序列，并将 δ-集和有序序列对应关系存储在预计算表中。在线阶段则需要猜测区分器前端和区分器后端的部分密钥，对选定的明密文进行加解密，并查看在预计算表是否有匹配，当匹配时则说明猜测的密钥可能是正确的密钥，否则是错误的。

2.4. 构造 MILP 模型

MILP 方法的主要思想是将寻找中间相遇区分器的问题转化为数学优化问题。BORON 主要包含三个操作：异或操作、分组置换操作和 S 盒操作。下面将给出这三个操作的约束条件。

异或操作 在 BORON 中会频繁出现异或操作，假设 $a,b,c\in\{0,1\}^4$，$a\oplus b=c$，使用不等式组(1)对异或操作进行约束：

$$\begin{cases} a+b+c\geq 2r \\ a\leq r, b\leq r, c\leq r \end{cases} \tag{1}$$

分组置换操作 在BORON中，分组置换层一般用来表示第 i 轮的输出和第 i+1 的输入之间的联系。使用不等式组(2)对分组置换层进行约束：

$$Y_i^j = X_{i+1}^{\pi}(j) \tag{2}$$

S 盒操作 假设 $m\times n$ 的 S 盒的其输入差分和输出差分分别记为 $(x_0,x_1,\cdots,x_{m-1}),(y_0,y,\cdots,y_{n-1})$，则可以用不等式组(3)来描述 S 盒的传播规则：

$$\begin{cases} A_t - x_i \geq 0, i\in\{0,1,\cdots,m-1\} \\ -A_t + \sum_{j=0}^{m-1} x_j \geq 0 \end{cases} \tag{3}$$

其中 $A_t\in\{0,1\}$ 是一个表示该 S 盒是否活跃的虚拟变量。当且仅当 $x_0,x_1,\cdots,x_{m-1}$ 不全为零时用 $A_t=1$，即代表该 S 盒为活跃 S 盒。$m+n$ 维的向量 Ω 表示 S 盒的输入输出差分模式，即 $\Omega=(x_0,x_1,\cdots,x_{m-1},y_0,y,\cdots,y_{n-1})\in F_2^{m+n}$，然后用 H 来表示所有的 S 盒可能的输入输出差分模式的凸包，并使用约减算法[13]对 H 进行去冗余操作，得到 H 的一个子集，将这个子集放入 SageMath 中，生成对应的不等式组。使用更少的不等式建立 MILP 模型来精确描述 S 盒的输入输出差分模式。

3. 基于 MILP 自动搜索 Boron 中间相遇区分器

BORON 的加密过程由 S 盒、分组置换、和轮密钥异或组成(异或操作的约束在之前已详细讨论)。本文的中间相遇区分器只在单密钥情况下使用。

S 盒 BORON 的 S 盒的差分分布表如表 2 所示，将表中元素表示为差分模式 $\Omega=(x_0,x_1,\cdots,x_{m-1},y_0,y,\cdots,y_{n-1})\in F_2^{m+n}$，并将差分模式放入 SageMath 工具中生成了 352 个不等式，再经过削减算法的去冗余后最终使用了不等式组(4)中的 23 个不等式即可描述 S_0 的差分性质，在此展示部分不等式，完整的 S 盒约束不等式见附录。因为 BORON 一共有 16 个 S 盒，因此需要 368 个不等式即可刻画所有 S 盒。

Table 2. The DDT of S box
表 2. S 盒的差分分布表

输入差分	输出差分
0000	0000
0001	0101, 0110, 1010, 1101, 1110, 1111
0010	0011, 0101, 1011, 1101
0011	0010, 0101, 0110, 1101, 1110, 1111
0100	0011, 0101, 0111, 1001, 1011, 1101
0101	0001, 0011, 0110, 0111, 1000, 1010, 1100, 1101
0110	0010, 0100, 1000, 1010, 1100, 1110
0111	0001, 0010, 0100, 0101, 0110, 0111, 1000, 1011

续表

1000	0011, 0110, 1011, 1100, 1110, 1111
1001	0001, 0010, 0100, 1001, 1010, 1100
1010	0011, 0100, 0110, 1011, 1110, 1111
1011	0001, 0010, 0100, 1001, 1010, 1100
1100	0001, 0101, 0110, 0111, 1000, 1010, 1011, 1100
1101	0010, 0011, 0100, 0101, 1010, 1011, 1100, 1101
1110	0001, 0010, 0011, 0100, 0110, 0111, 1000, 1101
1111	0111, 1000, 1001, 1110

$$\begin{cases} -x_3+x_1-x_0-y_2-y_1+3\geq 0 \\ x_3-3x_2+x_1-x_0-3y_3-2y_2-3y_1-y_0+10\geq 0 \\ -x_3-3x_2+x_1+x_0-3y_3-3y_2-2y_1-y_0+10\geq 0 \\ -x_3-2x_2+2x_1-x_0-y_3+2y_2+2y_1-y_0+4\geq 0 \\ \vdots \\ -2x_3-2x_2-x_1+2x_0-2y_3-y_2-y_1+y_0+7\geq 0 \\ 2x_3-2x_2-x_1-2x_0-2y_3-y_2-y_1+y_0+7\geq 0 \\ -2x_3-3x_2-3x_1+x_0-3y_3+y_2-2y_1-y_0+11\geq 0 \\ x_3-3x_2-3x_1-2x_0-3y_3-2y_2+y_1-y_0+11\geq 0 \end{cases} \tag{4}$$

分组置换层 假设 x_i, y_i 分别表示 BORON 置换层的输入和输出，则置换层不等式的建立如不等式组(5)所示：

$$\begin{cases} x_0-y_2=0 \\ x_1-y_3=0 \\ x_2-y_0=0 \\ x_3-y_1=0 \end{cases} \tag{5}$$

4. 9 轮 BORON 中间相遇攻击过程

这一章节主要介绍 5 轮中间相遇区分器的构造以及 9 轮中间相遇攻击的具体过程。

4.1. 5 轮 BORON 区分器的构造

在获得需要的不等式后，即可开始寻找中间相遇区分器。在算法 2 中使用 Gurobi 求解器求解模型 M，在保证至少有一个输入差分和一个输出差分是活跃的前提下，搜索到多条中间相遇区分器。表 3 仅列出部分 4 条中间相遇差分链，其中$(i;j,k)$表示区分器输入端活跃半字节位置为 X_i，输出端活跃半字节位置为 X_j, X_k。对表中的四条中间相遇差分链进行分析，发现差分链(4;4,5)需要用到的参数个数最少。图 2 详细展示了中间相遇差分链(4;4,5)的具体结构。

为了描述区分器，定义一个 δ -集 $\{M^0, M^1, \cdots, M^{15}\}$ 满足 $X_1[19\sim 16]$ 活跃，而 $X_1[63\sim 20, 15\sim 0]$ 非活跃，则经过 5 轮 Boron 加密后输出差分序列：$X_6^1[23\sim 16]\oplus X_6^0[23\sim 16]$，$X_6^2[23\sim 16]\oplus X_6^0[23\sim 16]$，$\cdots$，$X_6^{15}[23\sim 16]\oplus X_6^0[23\sim 16]$，可以由 $Y_1^0[19\sim 16]$，$Y_2^0[15\sim 12]$，$Y_2^0[31\sim 28]$，$Y_3^0[59\sim 56]$，$Y_3^0[55\sim 52]$，$Y_3^0[43\sim 40]$，$Y_3^0[39\sim 36]$，$Y_3^0[27\sim 24]$，$Y_3^0[11\sim 8]$，$Y_3^0[7\sim 4]$，$Y_4^0[63\sim 60]$，$Y_4^0[59\sim 56]$，$Y_4^0[55\sim 52]$，

$Y_4^0[51\sim48]$，$Y_4^0[47\sim44]$，$Y_4^0[43\sim40]$，$Y_4^0[39\sim36]$，$Y_4^0[35\sim32]$，$Y_4^0[31\sim28]$，$Y_4^0[27\sim24]$，$Y_4^0[15\sim12]$，$Y_4^0[11\sim8]$，$Y_4^0[7\sim28]$，$Y_4^0[27\sim24]$，$Y_4^0[15\sim12]$，$Y_4^0[11\sim8]$，$Y_4^0[7\sim4]$，$Y_4^0[3\sim0]$，$Y_5^0[63\sim60]$，$Y_5^0[55\sim52]$，$Y_5^0[51\sim48]$，$Y_5^0[27\sim24]$，$Y_5^0[23\sim20]$ 29 个参数表示，其中，$Y_r^0[i]=X_r^0[i]\oplus RK_r^0[i]$。即 $X_6[23\sim20], X_6[19\sim16]$ 处，30 个 4 比特差分序列仅有 $2^{29\times4}=2^{116}$ 种取值，而非 $2^{15\times2\times4}=2^{120}$ 种取值。

算法 2 自动化搜索 BORON 中间相遇区分器

输入：输入差分集合ΔI，输出差分集合ΔO，线性层和非线性层的不等式约束；

输出：*r* 轮的中间相遇区分器；

```
1)  初始化；
2)  for k=0 , k<r , k++ do
3)  输入非线性层约束条件；
4)  输入线性层约束条件；
5)  for i ∈ΔI do
6)      for j∈ΔO do
7)          增加限制条件：至少有一个输入差分和一个输出差分活跃
8)          构造精确的下限搜索模型 M
9)          使用 Gurobi 求解器求解模型 M
10)     end for
11) end for
12) return 中间相遇区分器
```

Table 3. The middle-meeting distinguisher
表 3. 搜索到的中间相遇区分器

区分器	区分器轮数	参数个数
(4;4,5)	5	29
(9;9,11)	5	31
(8;12,13)	5	30
(8;9,10)	5	30

证明：如图 2 所示，其中白色表示该半字节位置的差分为零，灰色表示此半字节位置的差分受区分器输入差分影响，黑色表示此半字节位置差分影响区分器的输出差分，同时受区分器输入差分影响。

假设区分器输入差分满足 $\Delta X_1[19\sim16]$ 为非零差分，在其他位置均为零差分。猜测 $Y_1^0[19\sim16]=X_1[19\sim16]\oplus RK_1[19\sim16]$，则第一轮 S 盒的输入差分已知，因为分组置换、移位操作、异或操作不会改变差分值，因此第一轮的输出差分已知；第二轮输入差分已知，猜测非零差分处的值 $Y_2^0[15\sim12], Y_2^0[31\sim28]$，则第二轮的输出差分已知；同样的，猜测非零差分处的值 $Y_3^0[59\sim56], Y_3^0[55\sim52], Y_3^0[43\sim40], Y_3^0[39\sim36], Y_3^0[27\sim24], Y_3^0[11\sim8], Y_3^0[7\sim4]$，则第三轮的输出差分已知；猜测非零差分处的值 $Y_4^0[63\sim60], Y_4^0[59\sim56], Y_4^0[55\sim52], Y_4^0[51\sim48], Y_4^0[47\sim44], Y_4^0[43\sim40], Y_4^0[39\sim36], Y_4^0[35\sim32], Y_4^0[31\sim28], Y_4^0[27\sim24], Y_4^0[15\sim12], Y_4^0[11\sim8], Y_4^0[7\sim4], Y_4^0[3\sim0]$，则第四轮的输出差分已知；猜测非零差分处的值 $Y_5^0[63\sim60], Y_5^0[55\sim52], Y_5^0[51\sim48], Y_5^0[27\sim24], Y_5^0[23\sim20]$ 则第五轮的输出差分已知，即 $X_6[23\sim20], X_6[19\sim16]$ 30 个 4 比特差分序列可以有上述 29 参数表示。

证毕。

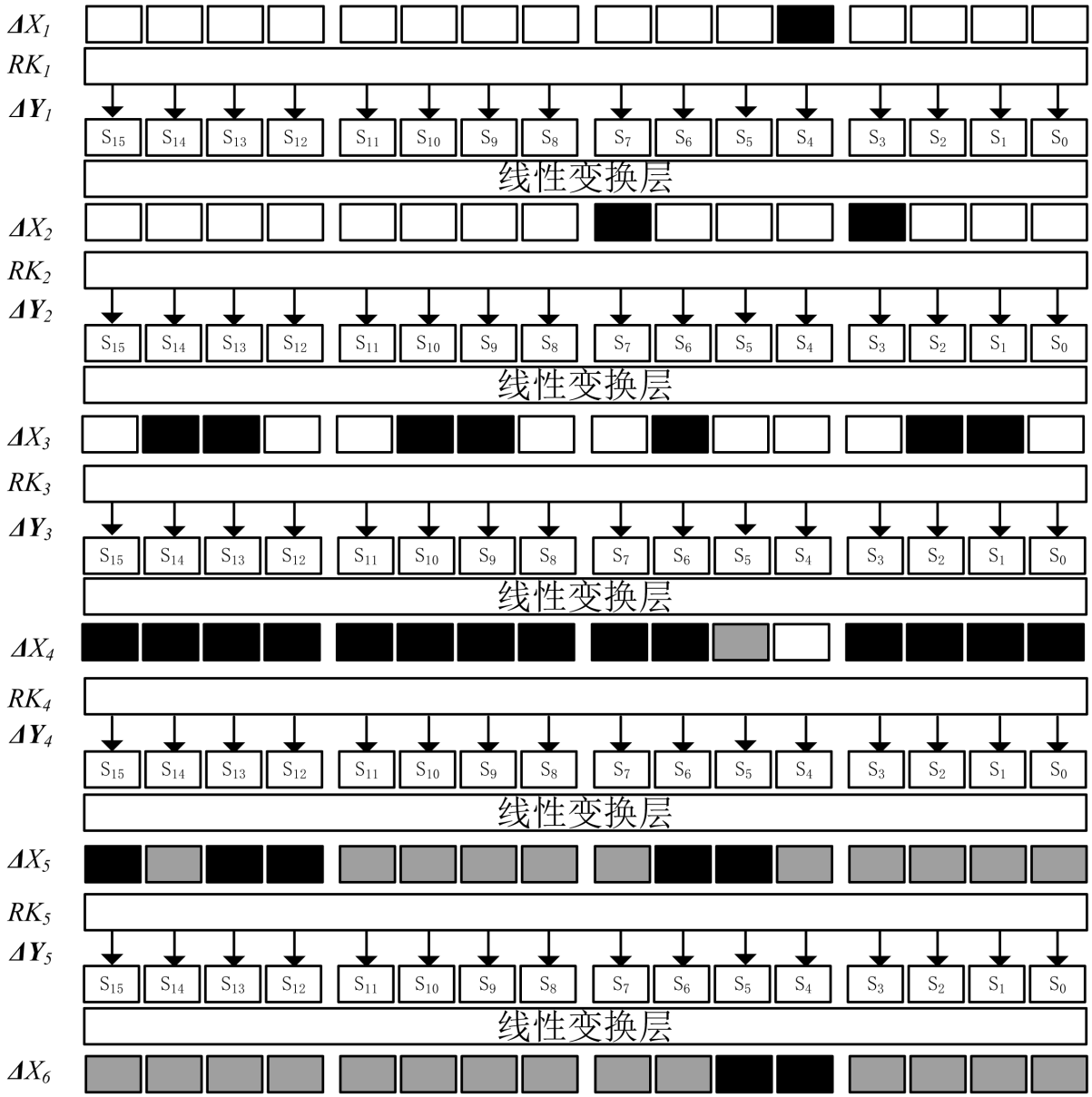

Figure 2. 5 rounds middle meeting distinguisher
图 2. 5 轮中间相遇区分器

4.2. 9 轮中间相遇攻击的具体过程

基于上述的 5 轮区分器，在前端加一轮，后面接三轮，对 9 轮 Boron-128 进行中间相遇攻击，攻击路径如图 3 所示。攻击分为在线和离线两个部分，具体如下。

4.2.1. 离线阶段

计算 120 比特差分序列的所有 2^{116} 种可能值，并存储于哈希表中。

4.2.2. 在线阶段

1) 选择明文 $M^0 = X_0^0$，猜测密钥 $RK_1[63\sim 60, 51\sim 48, 23\sim 20]$，部分加密 M^0 得到区分器的输入端 $X_1[19\sim 16]$。

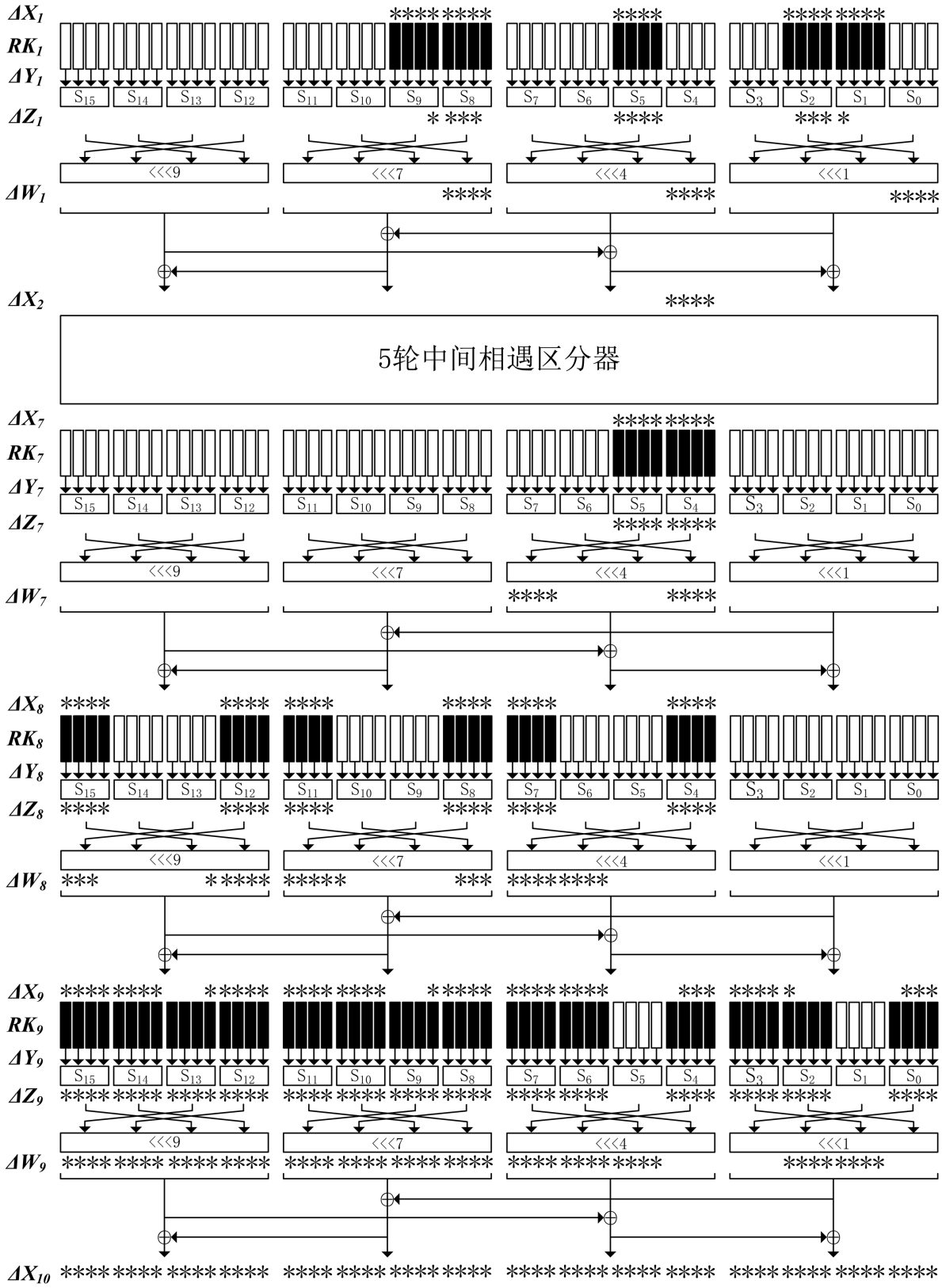

Figure 3. 9 round middle meeting attack
图 3. 9 轮中间相遇攻击示意图

2) 在区分器输入端构造 δ-集，令 $X_2[19\sim16]$ 遍历所有 16 种可能值，具体为令 $X_2^0[23\sim16]$ 异或 15 种非零差分，$X_2[63\sim20]$ 与 $X_2[15\sim0]$ 为未知常值。

3) 猜测状态 $Y_1^0[39\sim32,23\sim20,11\sim4]$，部分解密 δ-集，即可得到明文输入的活跃差分，与 $M^0=X_0^0$ 异或即可得符合 5 轮区分器特性的 16 个明文集合，并获得对应的密文。

4) 猜测密钥 $RK_9[63\sim56,55\sim40,39\sim24,19\sim8,3\sim0], RK_8[63\sim60,51\sim44,35\sim28,19\sim16], RK_7[23\sim16]$，部分解密密文得到 $X_2[23\sim16]$ 处差分序列。

5) 检查解密所得差分序列是否存在于离线阶段建立的哈希表中。若存在，则猜测密钥作为正确密钥候选值，否则删除错误密钥。在剩余密钥基础上穷搜恢复完整主密钥。

4.3. 复杂度分析

易知构造 δ-集时猜测的状态 $Y_1^0[39\sim32,23\sim20,11\sim4]$ 可在密钥 $RK_1[39\sim32,23\sim20,11\sim4]$ 下经过部分加密 M^0 得到，分析 Boron-128 的密钥关系发现，$RK_9[63\sim60]$ 可由 $RK_1[23\sim20]$ 推导而来，$RK_9[51\sim48]$ 可由 $RK_1[11\sim8]$ 推导而来，$RK_9[47\sim44]$ 可由 $RK_1[7\sim4]$ 推导而来，$RK_9[39\sim36]$ 可由 $RK_1[63\sim60]$ 推导而来，$RK_9[27\sim24]$ 可由 $RK_1[51\sim48]$ 推导而来，$RK_9[15\sim12]$ 可由 $RK_1[39\sim36]$ 推导而来，$RK_9[11\sim8]$ 可由 $RK_1[35\sim32]$ 推导而来，$RK_8[63\sim60]$ 可由 $RK_1[39\sim36], RK_1[35\sim32]$ 推导而来，$RK_8[35\sim32]$ 可由 $RK_1[11\sim8], RK_1[7\sim4]$ 推导而来，$RK_7[23\sim20]$ 可由 $RK_1[11\sim8], RK_1[7\sim4]$ 推导而来。因此在线阶段共猜测了 $28(RK_1)+8(RK_7)+24(RK_8)+56(RK_9)$-44 (可用密钥关系推导出的) = 72 比特密钥。

攻击所需数据量为 2^{20} 选择明文，离线阶段时间复杂度为 $2^4\times2^{116}$ 次部分加密，约为 $2^{116.84}$ 次 9 轮 Boron-128 加密，存储为 2^{116} 个 120 比特差分序列，约为 $2^{116.9}$ 个 64 比特块。在线阶段时间复杂度为 $2^{72}\times2^4$ 次部分加解密，约为 $2^{72.84}$ 次 9 轮加密。

为了使离线和在线阶段的复杂度达到平衡，采用时间–存储权衡(TMTO)策略，这意味着在离线阶段构建表格时，并不会保存所有 2^{116} 种可能的差分序列，而是选择按照特定的比例进行保存。假设仅存储 2^{116-m} 种差分序列，则离线阶段时间复杂度为 $2^{116.84-m}$ 次 9 轮加密，存储为 $2^{116.9-m}$ 个 64 比特块．在线阶段需重复 2^m 次来保证攻击的成功率，时间复杂度为 $2^{72.84+m}$ 次 9 轮加密，数据复杂度变为 2^{20+m} 选择明文。

选择 $m=22.00$，则攻击总的计算复杂度为 $2^{116.84-m}+2^{72.84+m}=2^{95.84}$ 次 9 轮加密，存储为 $2^{94.9}$ 个 64 比特块，数据复杂度为 2^{42} 选择明文。

5. 结论

本文研究了 BORON 抵抗中间相遇攻击的能力。实验表明对 9 轮 BORON-128 中间线相遇攻击所需的数据复杂度为 2^{42} 个选择明文，时间复杂度为 $2^{95.84}$ 次 9 轮 BORON 加密，存储复杂度为 $2^{94.9}$ 个 64 比特块。此结果是对 BORON 安全性分析的重要补充。

基金项目

国家自然科学基金项目(62002184)。

参考文献

[1] 杜小妮, 王香玉, 梁丽芳, 等. 轻量级分组密码 Piccolo 的量子密码分析[J]. 通信学报, 2023, 44(6): 175-182.

[2] 黄湘蜀, 王敏, 杜之波, 等. 针对轻量级分组密码算法 PRESENT 的随机差分故障攻击[J]. 成都信息工程大学学报, 2022, 37(1): 8-15. https://doi.org/10.16836/j.cnki.jcuit.2022.01.002

[3] Borghoff, J., Canteaut, A., Güneysu, T., *et al*. (2012) Prince—A Low-Latency Block Cipher for Pervasive Computing Applications. *Advances in Cryptology—ASIACRYPT* 2012: 18*th International Conference on the Theory and Application of Cryptology and Information Security*, Beijing, 2-6 December 2012, 208-225. https://doi.org/10.1007/978-3-642-34961-4_14

[4] 孟娇, 李濛, 胡晓波, 等. 轻量级 LBlock 算法硬件实现与研究[J]. 电力信息与通信技术, 2023, 21(3): 47-52. https://doi.org/10.16543/j.2095-641x.electric.power.ict.2023.03.07

[5] Beierle, C., Leander, G., Moradi, A., *et al*. (2019) Craft: Lightweight Tweakable Block Cipher Withefficient Protection against DFA Attacks. *IACRTransactions on Symmetric Cryptology*, **2019**, 5-45. https://doi.org/10.46586/tosc.v2019.i1.5-45

[6] Bansod, G., Pisharoty, N. and Patil, A. (2017) BORON: An Ultra-Lightweight and Low Power Encryption Design for Pervasive Computing. *Frontiers of Information Technology & Electronic Engineering*, **18**, 317-331. https://doi.org/10.1631/FITEE.1500415

[7] Liang, H. and Wang, M. (2019) Cryptanalysis of the Lightweight Block Cipher BORON. *Security and Communication Networks*, **2019**, 1-12. https://doi.org/10.1155/2019/7862738

[8] Li, Y., Liang, M., Lin, H., *et al*. (2020) Integral Attack on Reduced-Round BORON Based on Bit-Based Division Property. *Journal of Physics*: *Conference Series*, **1486**, Article ID: 022016. https://doi.org/10.1088/1742-6596/1486/2/022016

[9] Wu, X.N., Li, Y.X., Li, L.C., *et al*. (2022) Impossible Differential Cryptanalysis of BORON. *Journal of Information Science & Engineering*, **38**, 805-819.

[10] Diffie, W. and Hellman, M.E. (1977) Special Feature Exhaustive Cryptanalysis of the NBS Data Encryption Standard. *Computer*, **10**, 74-84. https://doi.org/10.1109/C-M.1977.217750

[11] Li, R. and Jin, C. (2016) Meet-in-the-Middle Attacks on 10-Round AES-256. *Designs*, *Codes and Cryptography*, **80**, 459-471. https://doi.org/10.1007/s10623-015-0113-3

[12] Sun, S., Hu, L., Wang, P., *et al*. (2014) Automatic Security Evaluation and (Related-Key) Differential Characteristic Search: Application to SIMON, PRESENT, LBlock, DES (L) and Other Bit-Oriented Block Ciphers. *Advances in Cryptology—ASIACRYPT* 2014: 20*th International Conference on the Theory and Application of Cryptology and Information Security*, Kaoshiung, 7-11 December 2014, 158-178. https://doi.org/10.1007/978-3-662-45611-8_9

[13] Sasaki, Y. and Todo, Y. (2017) New Algorithm for Modeling S-Box in MILP Based Differential and Division Trail Search. *Innovative Security Solutions for Information Technology and Communications: 10th International Conference*, *SecITC* 2017, Bucharest, 8-9 June 2017, 150-165. https://doi.org/10.1007/978-3-319-69284-5_11