密码学报 (中英文) ISSN 2097-4116 CN 10-1900/TN
Journal of Cryptologic Research, 2024, 12(4): 894–910

E-mail: jcr@cacrnet.org.cn
http://www.jcr.cacrnet.org.cn
Tel/Fax: +86-10-82789618

顺序编码方法优化与 SAT 搜索应用*

颜国华[1], 张凤荣[1], 崔 笑[1], 韦永壮[2], 王保仓[3]

1. 西安电子科技大学 网络与信息安全学院, 西安 710071
2. 桂林电子科技大学 计算机与信息安全学院, 桂林 541004
3. 西安电子科技大学 空天地一体化综合业务网全国重点实验室, 西安 710071

通信作者: 张凤荣, E-mail: zhangfengrong@xidian.edu.cn

摘 要: 自动搜索技术在密码分析中起着越来越重要的作用, SAT 搜索技术是目前常用的搜索技术之一. 为了更好地使得 SAT 搜索方法应用于密码分析领域并提高搜索效率, 本文提出新的构造 SAT 模型方法. 首先, 提出一种新的 k 输入异或模型, 在产生 $4\cdot(k-1)\cdot n$ 个句子情况下, 引入变量减少至 $\lceil\frac{(k-3)}{2}\rceil\cdot n$. 其次, 对约束目标函数的顺序编码方法改进, 提出了两种新的顺序编码方法, 两种方法引入的辅助变量分别减少至 $(n-\frac{k-1}{2})\cdot k$ 和 $(n-k)\cdot k$. 进一步地, 根据新的顺序编码方法提出新的定界条件编码方法, 将 Matsui 定界条件引入 SAT 模型加速搜索. 最后, 本文将新的构造 SAT 模型方法应用于 FBC、SMS4 和 PRESENT 等密码算法的最小活跃 S 盒搜索, 给出相应缩减轮密码算法的最小活跃 S 盒数量.

关键词: 分组密码; 差分分析; 自动搜索; SAT 方法

中图分类号: TP309.7 **文献标识码:** A DOI: 10.13868/j.cnki.jcr.000800

中文引用格式: 颜国华, 张凤荣, 崔笑, 韦永壮, 王保仓. 顺序编码方法优化与 SAT 搜索应用[J]. 密码学报 (中英文), 2024, 12(4): 894–910. [DOI: 10.13868/j.cnki.jcr.000800]

英文引用格式: YAN G H, ZHANG F R, CUI X, WEI Y Z, WANG B C. Optimization of sequential encoding method and SAT search application[J]. Journal of Cryptologic Research, 2024, 12(4): 894–910. [DOI: 10.13868/j.cnki.jcr.000800]

Optimization of Sequential Encoding Method and SAT Search Application

YAN Guo-Hua[1], ZHANG Feng-Rong[1], CUI Xiao[1], WEI Yong-Zhuang[2], WANG Bao-Cang[3]

1. School of Cyber Engineering, Xidian University, Xi'an 710071, China
2. School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China
3. State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China

Corresponding author: ZHANG Feng-Rong, E-mail: zhangfengrong@xidian.edu.cn

Abstract: Automated search techniques are becoming increasingly important in cryptanalysis. SAT search techniques are among the commonly used methods. To better apply SAT search methods in the field of cryptanalysis and improve search efficiency, this study proposes a new method for

* 基金项目: 国家自然科学基金面上项目 (62372346); 陕西高校青年创新团队
Foundation: General Program of National Natural Science Foundation of China (62372346); Youth Innovation Team of Shaanxi Universities
收稿日期: 2024-05-28 定稿日期: 2025-05-30

constructing SAT models. First, a new k-input XOR model is proposed, which reduces the number of variables to $\left\lceil \frac{(k-3)}{2} \right\rceil \cdot n$ while generating $4 \cdot (k-1) \cdot n$ clauses. Second, improvements are made to the sequential encoding method for constraint objective functions, introducing two new sequential encoding methods. The auxiliary variables introduced by these two methods are reduced to $\left(n - \frac{k-1}{2}\right) \cdot k$ and $(n-k) \cdot k$, respectively. Furthermore, based on the new sequential encoding methods, a new bounding condition encoding method is proposed, incorporating Matsui's bounding condition into the SAT model to accelerate the search. Finally, the new SAT model construction methods are applied to the search for the minimum active S-boxes in reduced-round ciphers of algorithms such as FBC, SMS4, and PRESENT, providing the minimum number of active S-boxes for these reduced-round ciphers.

Key words: block cipher; differential analysis; automatic search; SAT method

1 引言

分组密码在密码学领域占据着重要地位, 近年来, 人们不断提出新的设计密码方案, 旨在实现更安全、更高效的密码算法. Biham 和 Shamir 提出的差分密码分析技术是最流行的密码分析方法之一, 如今, 抵抗差分分析 [1] 和线性分析 [2] 被视为设计新密码算法的基本原则, 找到有效的差分和线性特征是执行这两种攻击的关键步骤.

在早期的密码分析中, 寻找有效的差分特征通常是通过手动搜索进行的, 但这种方法效率低. 2011 年, Mouha 等人提出了一种基于混合整数规划 (mixed-integer linear programming, MILP) 的自动搜索方法 [3], 利用该方法评估差分和线性活跃 S 盒下界, 从而使自动搜索方法开始在密码学领域中得到应用. Sun 等人 [4] 提出了一种可用于搜索差分特征的方法, 使自动搜索方法开始针对比特位密码算法进行应用. 随后, MILP 方法进一步在密码分析领域得到广泛应用, 应用于查找各种区分器, 例如不可能差分区分器 [5]、积分区分器 [6]、立方攻击 [7], 以及 ARX 密码的差分特征 [8] 等.

随着 MILP 的发展, 基于布尔可满足性问题 (Boolean satisfiability problem, SAT) 的自动搜索也开始应用于密码分析领域. 2013 年, Mouha 等人提出运用 SAT 方法 [9] 对 ARX 密码算法寻找差分特征. 随后, Kölbl 等人将其应用于 SIMON 类密码算法的差分搜索当中 [10]. 2021 年, Sun 等人 [11] 将 Matsui 的定界条件方法引入 SAT 建模当中, 加速 SAT 求解器的求解速度. 除此之外, 约束编程 (constraint programming, CP) 自动搜索方法也被应用于密码分析当中, 在文献 [12, 13] 中, 运用 CP 工具对 AES、SKINNY 等密码算法进行安全分析.

对于自动搜索方法, 用户可以通过简单的编写程序, 将密码算法的一些操作转化为底层的数学问题, 然后将构建好的模型运用相关的求解器进行处理即可. 然而, 目前对自动搜索效率并不是很理想, 特别是对一些 S 盒比特较大的密码算法, 搜索效率比较低. 因此, 目前对自动搜索方法的研究, 致力于提高搜索的效率. 文献 [4, 14] 认为最小化不等式的数量对于提高 MILP 的搜索效率很重要. 2017 年, Sasaki 等人提出新的算法, 使得构建 S 盒差分传播的不等式最小化, 以此提高搜索差分特征的效率 [15]. 2018 年, 文献 [16] 中将 Matsui 的定界条件 [17] 引入 MILP 搜索模型当中, 使得对部分算法的搜索效率得到提升.

为了使得 SAT 搜索的效率进一步提高, 应用于更多的密码分析领域, 本文致力于研究减少构建的 SAT 的模型当中的变量个数和产生的句子数量. 针对存在 k 个 n 比特向量输入异或操作的情况, 本文结合文献 [11] 中两种构造 SAT 模型方法的优势, 提出了 3-2 输入异或方法, 在产生 $4 \cdot (k-1) \cdot n$ 个句子情况下, 所需要的变量个数减少至 $\lceil \frac{(k-3)}{2} \rceil \cdot n$. 通过对顺序编码所对应的顺序计数电路进行研究, 发现顺序编码方法引入了过多的冗余变量. 对顺序编码方法进行了改进, 将一些固定值的辅助变量和不需要关注其值的辅助变量从顺序计数电路中删去, 提出了顺序编码方法 2 和顺序编码方法 3, 使得对目标函数约束产生的变量分别减少了 $\frac{(k-1) \cdot k}{2}$ 和 $(k-1) \cdot k$. 根据新的顺序编码方法辅助变量分配方案, 修改了文献 [11] 中的 Matsui 定界条件编码方法, 将 Matsui 定界条件引入 SAT 模型加速搜索. 同时还将新的构造 SAT 模型的方法应用于 SMS4、FBC 和 PRESENT 算法的最小活跃 S 盒搜索, 给出相应密码算法缩减轮最小活跃 S 盒数量.

本文内容安排如下: 第 2 节首先简单介绍 SAT 问题, 接着描述如何对分组密码算法构建 SAT 模型,

并提出新的异或方法; 第 3 节介绍顺序编码方法, 通过对顺序计数电路的研究提出新的顺序编码方法, 使得对目标函数构造 SAT 模型中需要的变量和句子数量减少; 第 4 节先简单介绍 Matsui 定界条件思想和构造 SAT 模型方法, 然后根据新的顺序编码方法引入辅助变量的策略, 提出新的定界条件编码方法; 第 5 节将本文构造 SAT 模型的方法应用于多个密码算法的最小活跃 S 盒搜索中; 第 6 节总结全文.

2 相关知识

2.1 SAT 问题

在介绍 SAT 问题之前, 首先介绍合取范式 (disjunctive normal form, CNF), 对于一个布尔公式, 其由布尔变量、运算符 AND ($\wedge$) OR ($\vee$) NOT ($\sim$) 和括号所组成, 而且每一个布尔公式其都可以转换为等价的合取范式的形式 (CNF) [18,19], 其可以写为 $\bigwedge_{i=0}^{n}\bigvee_{j=0}^{m_j} C_{ij}$, 其中 C_{ij} 被称为词, 由变量与常量所组成, 而 $\bigvee_{j=0}^{m_j} C_{ij}$ 被称为句子.

可满足性问题 SAT 源于数理逻辑中经典命题逻辑关于公式的可满足性的概念, 是理论计算机科学中一个重要的问题, 是确定二元变量是否存在评估的问题, 也是第一个被证明的 NP-complete [20] 问题, 对 SAT 问题的理论研究具有很多重要的意义.

对于大多数实际情况而言, SAT 问题的解决方案可以在合理的时间内找到. 目前存在着大量的启发式 SAT 求解器, 它们都接受 DIMACS CNF (合取范式) 文件作为标准输入格式. 在这些文件的格式中, 所有句子都是带有逻辑操作 OR 和 NOT 的变量, 句子与句子之间由 AND 操作连接. 当所构造的句子集合有解时, 输出 satisfiable, 否则输出 unsatisfiable. 对于有解的情况, 求解器还可以返回对所有变量的有效分配.

运用 SAT 自动搜索方法进行密码分析, 本质上就是利用一系列 CNF 等式将密码算法和目标函数描述出来, 然后交给求解器进行搜索, 判断是否有解.

2.2 SAT 模型的构建

如今的大多数对称密码算法都可以分为两大部分, 即线性部分和非线性部分. 这两个部分在密码算法的设计中起着至关重要的作用. 线性部分通常包括线性变换和线性混淆操作, 用于增加密码算法的复杂性和安全性. 而非线性部分则包括 S 盒、模块加等组件, 用于增加密码算法的混淆和扰乱性质.

掌握如何构建这两部分的 SAT 模型, 便可以对大多数分组密码算法构建 SAT 模型, 下面将分别介绍如何构建线性部分和非线性部分的 SAT 模型.

2.2.1 线性部分

对于分组密码的线性部分, 其差分传播是固定的, 线性部分主要由异或与移位等操作组成. 移位操作只需要在构造 SAT 模型时运用相应操作之后的变量, 因此, 对于线性部分而言, 关键在于如何将变量与变量之间的异或操作转换为 SAT 模型. 下文利用 α_i 来表示 n 比特向量 α 的第 i 比特.

差分模型 1 (两个变量向量间的异或) 对于两个 n 比特变量向量 α 和 β 之间的异或操作, 如图 1 所示, α 和 β 表示输入差分, γ 表示输出差分, 当且仅当 α、β 和 γ 同时满足式 (1) 时, 异或关系成立.

$$\left.\begin{array}{l}\overline{\alpha_i}\vee\beta_i\vee\gamma_i=1\\ \alpha_i\vee\overline{\beta_i}\vee\gamma_i=1\\ \overline{\alpha_i}\vee\overline{\beta_i}\vee\gamma_i=1\\ \alpha_i\vee\beta_i\vee\overline{\gamma_i}=1\end{array}\right\}0\le i\le n-1. \tag{1}$$

差分模型 2 (多输入异或操作) 对于 k 个输入差分的异或, 将这 k 个输入差分分别记为 $\alpha^0,\alpha^1,\cdots,\alpha^{k-1}$, 输出差分利用 β 来表示, 如图 2 所示. 在构建差分模型时, 需要对引入变量与句子数量进行平衡, 两者的过多引入都可能会导致求解器求解问题的时间增加.

文献 [11] 提出两种构建 SAT 模型的方法. 第一种方法是将这 k 个输入差分异或操作分解为 $(k-1)$ 个 2-输入的异或, 然后运用上面的差分模型 1 来表示 2-输入异或操作. 不过这需要引入额外的 $(k-2)\cdot n$ 个辅助变量来表示 $(k-2)$ 个中间状态, 其总共生成句子的数量为 $4\cdot(k-1)\cdot n$.

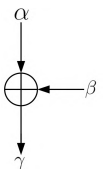

图 1 两个变量异或操作
Figure 1 XOR operation of two variables

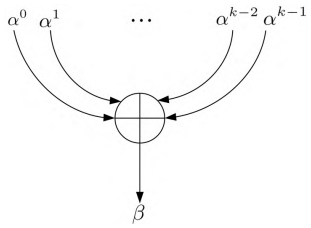

图 2 k 输入异或
Figure 2 k-input XOR

第二种方法便是在不引入辅助变量的情况下, 通过生成 $n\cdot 2^k$ 个 CNF 等式来表示 k 个输入的异或操作. 这是因为对于 $\left(\alpha_i^0,\alpha_i^1,\cdots,\alpha_i^{k-1},\beta_i\right)$ 这 $(k+1)$ 个变量所构成的向量一共有 2^{k+1} 个不同的取值. 对异或操作而言, 2^{k+1} 种不同的取值中有一半的取值不满足异或操作, 即存在 2^k 种取值不满足异或操作, 这 2^k 种不满足的异或的组合运用向量 $(\alpha_0,\alpha_1,\cdots,\alpha_k)$ 表示, 可以用式 (2) 将其排除.

$$\left(\alpha_i^0\oplus\alpha_0\right)\vee\left(\alpha_i^1\oplus\alpha_1\right)\vee\cdots\vee\left(\alpha_i^{k-1}\oplus\alpha_{k-1}\right)\vee\left(\alpha_i^k\oplus\alpha_k\right)=1,\quad 0\le i\le n-1. \tag{2}$$

在式 (2) 中, 当且仅当 $(\alpha_i^0,\alpha_i^1,\cdots,\alpha_i^{k-1},\beta_i)$ 为不满足异或的组合时, 等式不成立. 当向量 $(\alpha_0,\alpha_1,\cdots,\alpha_k)$ 中的 α_j 为 0 时, $\alpha_i^j\oplus\alpha_j$ 相当于 α_i^j, 当 α_j 为 1 时, $\alpha_i^j\oplus\alpha_j$ 可以运用 $\overline{\alpha_i^j}$ 表示.

通过研究上述两种 k 输入异或构建模型方法所产生的句子数量关系, 可以发现当 $k=3$ 时, 等式 (3) 成立, 两种方法产生的句子数量相同.

$$4\cdot(k-1)\cdot n=n\cdot 2^k. \tag{3}$$

因此, 对于 k 输入差分的异或, 本文提出新的构建 SAT 模型方法, 将该方法记为 3-2 输入异或方法. 对于 k 输入差分异或的情况, 首先将其分解为 3 个输入的异或模型, 通过引入中间变量完成 k 输入异或操作, 如图 3 所示. 若是分解到最后, 差分向量只剩一个, 则运用差分模型 1 来构造. 运用这种方法生成的句子数量与分解为 2 个输入异或生成的句子数量相同, 但是引入的辅助变量为 $\lceil\frac{(k-3)}{2}\rceil\cdot n$ 个, 少于分解为 2 输入异或方法引入的辅助变量.

差分模型 3 (变量与常数的异或) 对于多个变量向量与常数向量的异或, 下面以两个变量向量与一个常数向量相异或为例. 将变量向量记为 α,β, 常数向量记为 c, 输出记为 γ, 如图 4 所示.

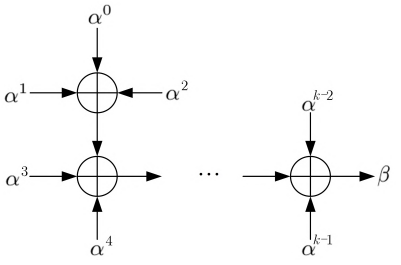

图 3 3-2 输入异或
Figure 3 3-2 input XOR

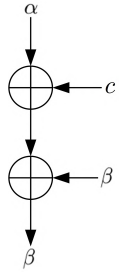

图 4 变量与常数的异或
Figure 4 XOR of variables and constants

将常数向量 c 中比特位值为 1 的序列号记录在列表 L 中, 在布尔表达式中, 异或上常数 1 相当于将输出值取反, 因此对于序列号 j 在列表 L 中的比特位异或, 当且仅当式 (4) 全部成立时, 异或关系成立.

$$\left.\begin{array}{l}\alpha_j\vee\beta_j\vee\gamma_j=1\\ \alpha_j\vee\overline{\beta_j}\vee\overline{\gamma_j}=1\\ \overline{\alpha_j}\vee\beta_j\vee\overline{\gamma_j}=1\\ \overline{\alpha_j}\vee\overline{\beta_j}\vee\gamma_j=1\end{array}\right\}j\in L. \tag{4}$$

而对于不在 L 中的序列号, 其相当于异或上 0, 输出值保持不变, 因此可以运用上述的差分模型 1 来表示. 这种类型的差分模型可以应用于内部差分技术的搜索当中, 在文献 [21] 对 SHA3 进行的内部差分搜索当中, 便存在着变量与常数的异或情况.

2.2.2 S 盒模型构建

对于分组密码中的非线性部分本文只介绍最常见的 S 盒模型, 模块加法操作和 SIMON 类轮函数的操作可以参考文献 [9,10]. 对于非线性部分, 其差分的传播是概率性的, 受进入非线性部分的具体值影响.

以 S 盒为组件的密码算法, 运用自动化搜索工具对密码进行差分分析主要为了实现两个目标. 一是找到最小活跃 S 盒的差分特征, 二是寻找最大差分概率的差分特征. 以下将以搜索最小活跃 S 盒的差分特征为例进行说明, 而寻找最大差分概率的方法, 与搜索最小活跃 S 盒的思路大致相近.

记 $(\alpha_0,\alpha_1,\cdots,\alpha_{n-1})$ 和 $(\beta_0,\beta_1,\cdots,\beta_{n-1})$ 分别为 n 比特 S 盒的输入和输出差分, 如图 5 所示, 并且引入辅助变量 w 来表示 S 盒是否活跃, 则构造 S 盒模型具体步骤如下:

图 5 S 盒模型

Figure 5 S-box model

第一步: 构建 S 盒的差分分布表. 根据差分分布表的值, 对于概率大于 0 而小于 1 的差分传播 $(\alpha_0,\alpha_1,\cdots,\alpha_{n-1})\to(\beta_0,\beta_1,\cdots,\beta_{n-1})$, 设置为 $w=1$ 表示 S 盒活跃. 而对于差分概率为 1 的差分传播, 将 w 设置为 0 表示 S 盒非活跃.

第二步: 根据差分分布表, 构造出所有 $(\alpha_0,\alpha_1,\cdots,\alpha_{n-1},\beta_0,\beta_1,\cdots,\beta_{n-1},w)$ 不可能传播组合. 不可能传播组合由两部分组成, 第一部分是使得 S 盒活跃的差分对但 $w=0$ 的组合和使 S 盒不活跃的差分对但 $w=1$ 的组合, 第二部分为不可能传播的差分对和 w 的组合. 假设差分对 $(\alpha_0^j,\alpha_1^j,\cdots,\alpha_{n-1}^j)\to(\beta_0^j,\beta_1^j,\cdots,\beta_{n-1}^j)$ 是可以使 S 盒活跃的传播, 其数量为 m, 令 w 的值为 0, 则 $(\alpha_0^j,\alpha_1^j,\cdots,\alpha_{n-1}^j,\beta_0^j,\beta_1^j,\cdots,\beta_{n-1}^j,0)$ 便是不可能传播组合, 其中 $\alpha_i^j,\beta_i^j\in\{0,1\}$, 当公式 (5) 成立时将不可能组合 $(\alpha_0^j,\alpha_1^j,\cdots,\alpha_{n-1}^j,\beta_0^j,\beta_1^j,\cdots,\beta_{n-1}^j,0)$ 排除在外, 当且仅当 $\alpha_i=\alpha_i^j,\beta_i=\beta_i^j,w=0$ 时, 等式左边才为 0.

$$\bigvee_{i=0}^{n-1}(\alpha_i\oplus\alpha_i^j)\vee\bigvee_{i=0}^{n-1}(\beta_i\oplus\beta_i^j)\vee(w\oplus 0)=1,0\le i\le m-1. \tag{5}$$

现在考虑第二种组合, 当不可能差分对的数量为 η 个时, 由于 w 可以取 0 和 1, 因此第二种类型的组合一共有 2η 种, 将 2η 种组合写成等式 (6), 便可以将第二种类型的所有组合排除在外.

$$\bigvee_{i=0}^{n-1}(\alpha_i\oplus\alpha_i^j)\vee\bigvee_{i=0}^{n-1}(\beta_i\oplus\beta_i^j)\vee(w\oplus w_i)=1,0\le i\le 2\eta-1. \tag{6}$$

但这样的直接构造, 会使得模型中的句子的数量比较多. 在这些句子的集合中, 可以将多个句子化简为一个句子. 为了减少句子的数量, 需要进行下面第三步操作.

第三步: 为了较少构造 S 盒模型的句子, 首先定义一个函数 $f(x)$, 其中 x 为 $2n+1$ 比特的向量, $x=(x_0,x_1,\cdots,x_{2n})$,

$$f(x)=\begin{cases}0, & x\text{ 是不可能组合}\\ 1, & \text{其他}\end{cases},$$

则 $f(x)$ 可以等价表示为以下乘积和的式子 [22]:

$$f(x)=\bigwedge_{c\in F_2^{2n+1}}\left(f(c)\vee\bigvee_{i=0}^{2n}(x_i\oplus c_i)\right),$$

其中, $c=(c_0,c_1,\cdots,c_{2n})$. 转换为乘积和的形式后, 便将减少句子数量的问题变成了简化布尔函数乘积

和表示的问题, 这类问题可以运用 Quine-McCluskey (QM) 算法或者通过一些现成的软件来实现, 例如 Logic Friday 和 Espresso 等工具.

Logic Friday 软件是一款具有图形化界面的逻辑综合工具, 操作简便, 适用于 Windows 平台. 然而, 其最大的局限性在于只能处理最多 16 个变量的逻辑化简. 当需要处理的变量个数超过 16 时, 可以借助 Espresso 工具进行逻辑化简, 支持化简更多变量个数.

2.3 目标函数与额外约束

目标函数是指构造 SAT 模型目的, 运用自动工具化搜索差分特征, 常见的目的要么是搜索最小活跃 S 数量差分特征, 要么是搜索最大差分概率的差分特征. 对于搜索最小活跃 S 盒的差分特征, 目标函数便是判定活跃 S 盒的个数总和是否达到一个指定值.

为了避免输入差分全为 0, 需要对第一轮的输入差分值添加额外的约束条件, 使得第一轮的输入差分值至少存在 1 比特的布尔变量的值为 1, 假设分组的大小为 m 比特, 则可以添加下列 CNF 等式 (7) 使得第一轮输入差分中至少存在 1 比特的布尔变量的值为 1.

$$x_0 \vee x_1 \vee \cdots \vee x_{m-1} = 1. \tag{7}$$

3 顺序编码方法的优化

3.1 顺序编码方法

在差分分析中, 最关键的便是找到高概率的差分区分器, 使得差分攻击能有效破解密钥. 根据具体目标, 可以运用自动化工具搜索时约束活跃 S 盒的数量或者差分概率的差分区分器, 这些约束条件都可以转换为布尔基数约束 $\sum_{j=0}^{n-1} x_j \leq k$, 其中 x_j 表示为布尔变量, k 为一个大于或等于 0 的整数. 在文献 [11, 22, 23] 中, 为了将布尔基数约束不等式转换为 CNF 公式, 采用顺序编码方法 [24] 将此约束转换为 CNF 公式.

该顺序编码方法是基于图 6 顺序计数电路图设计的, 采用一元编码系统来计算部分和 $s_i = \sum_{j=0}^{i} x_j$ $(0 \leq i \leq n-2)$. 为了将顺序计数电路图转换为 CNF 公式, 需要引入 $(n-1)\cdot k$ 个辅助变量 $s_{i,j}$ $(0 \leq i \leq n-2,\ 0 \leq j \leq k-1)$, 部分和 s_i 在一元编码系统下运用 $s_{i,k-1} \| s_{i,k-2} \| \cdots \| s_{i,0}$ 表示, 当 $s_{i,j}$ $(0 \leq j \leq k-1)$ 中有多少个 1 时, 部分和 s_i 便等于多少. 当 $s_i = m$ $(0 \leq m \leq k)$ 时, 可以用下式表示:

$$\underbrace{0\|\cdots\|0}_{k-m}\ \underbrace{\|1\|\cdots\|1}_{m}.$$

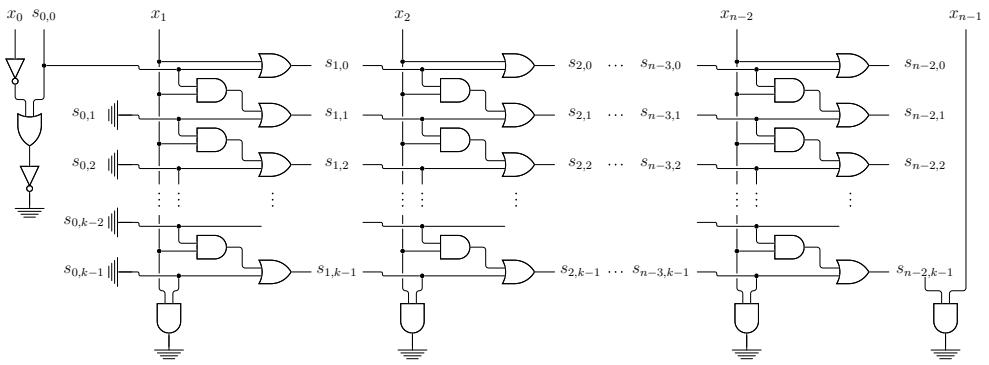

图 6 顺序计数器电路

Figure 6 Sequential counter circuit

因此, 等式 $s_i = \sum_{j=0}^{i} x_j$ 可以等价转换为 $s_i = \sum_{j=0}^{k-1} s_{i,j}$. 所以, 可以运用 $k\cdot(2n-3)+n-1$ 个句

子将布尔基数约束转换为 SAT 模型. 当且仅当式 (8) 全部成立时, 约束条件 $\sum_{j=0}^{n-1} x_j \leq k$ 成立.

$$\left.\begin{array}{l}
\overline{x_0} \vee s_{0,0} = 1 \\
\overline{s_{0,j}} = 1, 1 \leq j \leq k-1 \\
\left.\begin{array}{l}
\overline{x_i} \vee s_{i,0} = 1 \\
\overline{s_{i-1,0}} \vee s_{i,0} = 1 \\
\left.\begin{array}{l}
\overline{x_i} \vee \overline{s_{i-1,j-1}} \vee s_{i,j} = 1 \\
\overline{s_{i-1,j}} \vee s_{i,j} = 1
\end{array}\right\} 1 \leq j \leq k-1 \\
\overline{x_i} \vee \overline{s_{i-1,k-1}} = 1
\end{array}\right\} 1 \leq i \leq n-2 \\
\overline{x_{n-1}} \vee \overline{s_{n-2,k-1}} = 1
\end{array}\right. \tag{8}$$

3.2 顺序编码方法 2

在顺序编码方法当中, 对每一个布尔变量 $x_i\,(0 \leq i \leq n-2)$, 都相应地引入 k 个辅助变量 $s_{i,j}\,(0 \leq j \leq k-1)$, 用来计算部分和 s_i. 而通过对图 6 顺序计数电路的研究, 发现对于前 $k-1$ 个布尔变量 $x_i\,(0 \leq i \leq k-2)$ 相加, 部分和 $s_i\,(0 \leq i \leq k-2)$ 的值必然小于 k, 即 $\sum_{i=0}^{k-2} x_i < k$. 所以, 对于前 $k-1$ 个布尔变量而言, 引入了多余的辅助变量. 因为当 $s_{i,m} = 0$ 时, $s_{i,j}\,(m < j \leq k-1)$ 也必然为 0.

在顺序计数电路中, 当计算到布尔变量 x_1 时, 部分和 s_1 最大为 2, 所以 x_1 所对应的辅助变量 $s_{1,j}$ 最多只有 $s_{1,0}$ 和 $s_{1,1}$ 的值为 1, 后 $(k-2)$ 个 $s_{1,j}$ 的值是确定的, 它们的值为 0. 当计算到布尔变量 x_2 时, 部分和 s_2 最大为 3, x_2 所对应的 $s_{2,j}$ 最多只有 $s_{2,0}$、$s_{2,1}$ 和 $s_{2,3}$ 的值为 1, 而后 $(k-3)$ 个 $s_{2,j}$ 的值也只能为 0. 以此类推, 对于前 $k-1$ 个 x_i 所对应的 $s_{i,j}$ 皆是如此, 可以将顺序计数电路中辅助变量值确定的删去, 对每一个 x_i 重新分配辅助变量, 如图 7 所示.

$$\begin{array}{ccccccc}
x_0 & x_1 & \cdots & x_{k-1} & \cdots & x_{n-2} & x_{n-1} \\
S_{0,0} & S_{1,0} & \cdots & S_{k-1,0} & \cdots & S_{n-2,0} & \\
 & S_{1,1} & \cdots & S_{k-1,1} & \cdots & S_{n-2,1} & \\
 & & \ddots & \vdots & \cdots & \vdots & \\
 & & & S_{k-1,k-1} & \cdots & S_{n-2,k-1} &
\end{array}$$

图 7 顺序编码方法 2 变量示意图

Figure 7 Variable schematic diagram of sequential encoding method 2

由于辅助变量分配的不同, 为了将布尔基数约束 $\sum_{j=0}^{n-1} x_j \leq k$ 转换为 SAT 模型, 需要构造新的编码方法. 我们提出了下面的顺序编码方法 2 , 在顺序编码方法 2 中, 对前 $k-1$ 个布尔变量所对应的部分和表达式进行修改, 对于后面的布尔变量所对应的部分和采用原顺序编码方法. 当且仅当公式 (9) 全部成立时, 约束条件 $\sum_{j=0}^{n-1} x_j \leq k$ 成立.

$$\left.\begin{array}{l}
\overline{x_0} \vee s_{0,0} = 1 \\
\left.\begin{array}{l}
\overline{x_i} \vee s_{i,0} = 1 \\
\overline{s_{i-1,0}} \vee s_{i,0} = 1
\end{array}\right\} 1 \leq i \leq n-2 \\
\left.\begin{array}{l}
\left.\begin{array}{l}
\overline{x_i} \vee \overline{s_{i-1,j-1}} \vee s_{i,j} = 1 \\
\overline{s_{i-1,j}} \vee s_{i,j} = 1
\end{array}\right\} 1 \leq j \leq i-1 \\
\overline{x_i} \vee \overline{s_{i-1,i-1}} \vee s_{i,i} = 1
\end{array}\right\} 1 \leq i \leq k-1 \\
\left.\begin{array}{l}
\overline{x_i} \vee \overline{s_{i-1,j-1}} \vee s_{i,j} = 1 \\
\overline{s_{i-1,j}} \vee s_{i,j} = 1
\end{array}\right\} 1 \leq j \leq k-1, k \leq i \leq n-2 \\
\overline{x_i} \vee \overline{s_{i-1,k-1}} = 1, k \leq i \leq n-2 \\
\overline{x_{n-1}} \vee \overline{s_{n-2,k-1}} = 1
\end{array}\right. \tag{9}$$

在顺序编码方法 2 中, 通过删去固定值的辅助变量, 使得引入的辅助变量相较于顺序编码方法减少了 $\frac{(k-1)\cdot k}{2}$, 需要的变量个数变为 $\left(n-\frac{k-1}{2}\right)\cdot k$, 约束目标函数所需句子数量也相应的减少.

3.3 顺序编码方法 3

除了发现前 $k-1$ 个布尔变量的部分和 $\sum_{i=0}^{k-2} x_i < k$ 外, 仍可以发现在顺序编码方法中, 布尔变量 x_{n-1} 没有引入辅助变量, x_{n-1} 只需要判断部分和 s_{n-2} 是否为 k 即可. 在一元编码系统中, 判断部分和 s_{n-2} 是否为 k, 只需判断 $s_{n-2,k-1}$ 是否为 1. 所以, 在顺序编码方法中, 引入了等式 (10), 当且仅当 $x_{n-1}=1$ 和 $s_{n-2,k-1}=1$ 时, 等式 (10) 不成立.

$$\overline{x_{n-1}} \vee \overline{s_{n-2,k-1}} = 1. \tag{10}$$

受到上述思想的启发, 可以发现顺序编码中后 $(k-1)$ 个布尔变量 x_i 都是如此. 对于布尔变量 x_{n-2} 而言, 需要关注的可以分为两部分, 一是 x_{n-3} 所对的部分和是否为 k, 二是决定辅助变量 $s_{n-2,k-1}$ 值的辅助变量. 对于第一部分可以等价转换为判断辅助变量 $s_{n-3,k-1}$ 是否为 1. 而对于第二部分, 通过顺序编码方法中的 CNF 表达式 $\overline{x_i} \vee \overline{s_{i-1,j-1}} \vee s_{i,j}=1$ 和 $\overline{s_{i-1,j}} \vee s_{i,j}=1$ 可以知道, $s_{n-2,k-1}$ 的值取决于辅助变量 $s_{n-3,k-1}$ 和 $s_{n-3,k-2}$, 所以只需要关注辅助变量 $s_{n-3,k-1}$ 和 $s_{n-3,k-2}$ 取值即可, 而对于辅助变量 $s_{n-3,j}\,(0 \le j \le k-3)$ 的取值不需要关注. 对于布尔变量 x_{n-3} 而言, 需要关注的也为两部分, 一是 x_{n-4} 所对应的部分和是否为 k, 二是决定辅助变量 $s_{n-3,k-1}$ 和 $s_{n-3,k-2}$ 值的辅助变量. 通过上面类似的推导, 可以知道需要关注辅助变量只有 $s_{n-4,k-1}$、$s_{n-4,k-2}$ 和 $s_{n-4,k-3}$ 的值.

以此类推, 对于后面的 $(k-1)$ 个布尔变量 x_i 皆是如此. 由于对这些冗余辅助变量取值不需要关注, 因此可以将其从顺序计数电路中删去, 对后面的 $(k-1)$ 个布尔变量重新分配辅助变量, 如图 8 所示.

由于辅助变量的减少, 需要构造新的顺序编码方法将布尔基数约束 $\sum_{j=0}^{n-1} x_j \le k$ 转换为 SAT 模型. 我们提出下列顺序编码方法 3, 在新的顺序编码方法中, 对后 $(k-1)$ 布尔变量 x_i 的部分和进行修改, 而对于前面的布尔变量仍然采用顺序编码方法 2 的编码方法. 当且仅当公式 (11) 全部成立时, 约束条件 $\sum_{j=0}^{n-1} x_j \le k$ 成立.

$$\begin{aligned}
&\overline{x_0} \vee s_{0,0} = 1\\
&\left.\begin{aligned}&\overline{x_i} \vee s_{i,0} = 1\\ &\overline{s_{i-1,0}} \vee s_{i,0} = 1\end{aligned}\right\} 1 \le i \le n-k-1\\
&\left.\begin{aligned}&\left.\begin{aligned}&\overline{x_i} \vee \overline{s_{i-1,j-1}} \vee s_{i,j} = 1\\ &\overline{s_{i-1,j}} \vee s_{i,j} = 1\end{aligned}\right\} 1 \le j \le i-1\\ &\overline{x_i} \vee \overline{s_{i-1,i-1}} \vee s_{i,i} = 1\end{aligned}\right\} 1 \le i \le k-1\\
&\left.\begin{aligned}&\overline{x_i} \vee \overline{s_{i-1,j-1}} \vee s_{i,j} = 1\\ &\overline{s_{i-1,j}} \vee s_{i,j} = 1\end{aligned}\right\} 1 \le j \le k-1, k \le i \le n-k-1\\
&\left.\begin{aligned}&\overline{x_i} \vee \overline{s_{i-1,j-1}} \vee s_{i,j} = 1\\ &\overline{s_{i-1,j}} \vee s_{i,j} = 1\end{aligned}\right\} i-n-k+1 \le j \le k-1, n-k \le i \le n-2\\
&\overline{x_i} \vee \overline{s_{i-1,k-1}} = 1, \quad k \le i \le n-2\\
&\overline{x_{n-1}} \vee \overline{s_{n-2,k-1}} = 1
\end{aligned} \tag{11}$$

对于顺序编码方法 3, 将顺序计数电路中不需要关注的辅助变量删去, 在顺序编码方法 2 的基础上进一步将引入的辅助变量减少了 $\frac{(k-1)\cdot k}{2}$, 需要的辅助变量变为了 $(n-k)\cdot k$, 产生的句子数量也进一步减少.

$$\begin{array}{cccccccccc}
x_0 & x_1 & \cdots & x_{k-1} & \cdots & x_{n-k-1} & x_{n-k} & \cdots & x_{n-2} & x_{n-1} \\
S_{0,0} & S_{1,0} & \cdots & S_{k-1,0} & \cdots & S_{n-k-1,0} & & & & \\
 & S_{1,1} & \cdots & S_{k-1,1} & \cdots & S_{n-k-1,1} & S_{n-k,1} & & & \\
 & & \ddots & \vdots & \cdots & \vdots & \vdots & \ddots & & \\
 & & & S_{k-1,k-1} & \cdots & S_{n-k-1,k-1} & S_{n-k,k-1} & \cdots & S_{n-2,k-1} &
\end{array}$$

图 8 顺序编码方法 3 变量示意图

Figure 8 Variable schematic diagram of sequential encoding method 3

4 Matsui 定界条件

1994 年, Matsui 提出了分支定界深度优先搜索算法, 通过运用已知路径的上限概率信息, 来提高最优差分路径的搜索效率. 文献 [11] 通过借鉴文献 [16] 的思路, 将定界条件转换为 SAT 模型, 以此提高搜索差分特征的速度. 令 $N_A(i)$ 表示为 $i\,(1 \le i \le R-1)$ 轮活跃 S 盒数量的下界, 当完成对前 $R-1$ 轮的最小活跃 S 盒搜索时, 想要搜索 R 轮的最小活跃 S 盒数量. 设 $N_{\mathrm{Ini}}(R)$ 为我们对 R 轮最小活跃 S 盒的初始估计, 根据 Matsui 所提出的分支定界深度优先搜索算法, 当式 (12) 不成立时, $N_{\mathrm{Ini}}(R)$ 必然不是 R 轮活跃 S 盒的下界.

$$N_A(R-r) + N_A(r) \le N_{\mathrm{Ini}}(R). \tag{12}$$

为了寻找到一条活跃 S 盒不大于 $N_{\mathrm{Ini}}(R)$ 的差分特征 $\left(\alpha^0, \alpha^1, \cdots, \alpha^R\right)$, 需要使得式 (13) 成立

$$\sum_{i=0}^{R-1} \left(N_A\left(\alpha^i \to \alpha^{i+1}\right)\right) \le m_{\mathrm{Ini}}(R). \tag{13}$$

假设一轮有 m 个 S 盒, 通过引入布尔变量 $w_j^i\,(0 \le j \le m-1)$ 来计算第 i 轮差分特征 $\left(\alpha^i \to \alpha^{i+1}\right)$ 中活跃 S 盒的数量, 因此 $N_A\left(\alpha^i \to \alpha^{i+1}\right) = \sum_{j=0}^{m-1} w_j^i$. 令 $n = R \cdot m$, $k = N_{\mathrm{Ini}}(R)$, $x_{m\cdot i+j} = w_j^i$, 那么式 (13) 可以重新写为下式

$$\sum_{i=0}^{n-1} x_i = \sum_{i=0}^{R-1} \sum_{j=0}^{m-1} w_j^i \le k. \tag{14}$$

相应地, 式 (12) 可以等价地写为

$$\sum_{i=0}^{r-1} \left(N_A\left(\alpha^i \to \alpha^{i+1}\right)\right) \le N_{\mathrm{Ini}}(R) - N_A(R-r). \tag{15}$$

不等式 (15) 的右端可以运用一个常数 m 进行表示, 该值不大于 $m_{\mathrm{Ini}}(R)$, 不等式的左边表示的是前 r 轮 S 盒变量相加. 因此, 不等式 (15) 还可以进一步推广, 写为不等式 (16), 将不等式 (16) 记为第二个约束条件, 布尔基数约束 $\sum_{j=0}^{n-1} x_j \le k$ 记为第一个约束条件.

$$\sum_{i=e_1}^{e_2} x_i \le m. \tag{16}$$

4.1 定界条件编码

为了将第二个约束条件引入 SAT 模型, 而不新增辅助变量, 文献 [11] 利用第一个约束条件中引入的辅助变量, 根据式 (16) 中 e_1, e_2 的取值, 提出了三种类型 CNF 等式编码方式, 将第二个约束条件转换为 SAT 模型, 下面介绍常用的类型一编码.

当 $e_1 = 1, e_2 < n-1$ 时, 为了满足 $\sum_{i=e_1}^{e_2} x_i \le m$ 约束条件, 添加下列额外的 CNF 等式, 可以使得第二个约束同时成立. 因为在一元编码系统中 $\sum_{j=0}^{i} x_j$ 和 $\sum_{j=0}^{k-1} s_{i,j}$ 具有等价关系, 当 $s_{i,j} = 0$ 时, $s_{i,j'}\,(j < j' \le k-1)$ 也必然为 0.

$$\overline{x_i} \vee \overline{s_{i-1,m-1}} = 1, 1 \leq i \leq e_2. \tag{17}$$

4.2 新的定界条件编码

对于上述所介绍的边界编码方法, 是在原先的顺序计数电路中将边界约束条件转换 SAT 模型的. 而由于前文对顺序编码方法进行了优化, 分别将一些确定的辅助变量和不关心的辅助变量从顺序计数电路中删去, 使得所引入的辅助变量和产生的句子数量减少. 因此, 上述的定界条件编码 (17) 不再适用于新的顺序编码方法, 为了将定界条件转换为 SAT 模型, 需要提出新的定界条件编码方法. 当然, 也可以通过引入辅助变量的方法将定界条件转换为 SAT 模型, 但这会使得整体的变量和句子数量增加. 通过借鉴文献 [11] 的编码方法, 提出新的定界条件编码方法 (18).

对于新的定界条件编码, 与定界条件编码 (17) 不同之处主要在于 i 的取值范围. 在新的顺序编码方法中, x_1 所对应的 $s_{1,j}$ 辅助变量只有 $s_{1,0}, s_{1,1}$ 两个, 当 $m > 2$ 时, 上述定界条件编码中需要的辅助变量 $s_{1,m-1}$ 不存在. 当 $i \geq m-1$ 时, 所对应的辅助变量 $s_{i,m-1}$ 才存在. 换一个角度思考, 既然没有引入相应的辅助变量, 便不需要对相应的辅助变量进行约束, 所以定界条件编码的下界从 1 变为 m 即可. 对于定界条件编码中 i 的上界, 仍然可以取到 e_2. 因为 R 轮的最小活跃 S 盒数量与 $R-1$ 轮的最小活跃 S 盒数量之差, 必然不会大于轮函数中 S 盒的数量, 也就是说对于布尔变量 x_{e_2-1}, 所对应的辅助变量 s_{e_2-1,e_2-1} 必然存在.

$$\overline{x_i} \vee \overline{s_{i-1,m-1}} = 1, m \leq i \leq e_2. \tag{18}$$

5 实验

本节将构建 SAT 模型的新方法应用于对多个密码算法进行最小活跃 S 盒的搜索. 需要说明的是, 这一部分的所有实验数据都是在配备 13th Gen Intel(R) Core(TM) i5-13400F 的个人台式机上运行所得. 采用 Python 代码生成密码算法的 SAT 模型, 然后利用 SAT 求解器 CaDiCaL [25] 对生成的 SAT 模型文件判断是否有解 [1], 对于各个实验的详细搜索时间对比见附录.

5.1 k 输入异或方法对比

FBC 是由冯秀涛等人所设计的分组密码算法 [26], 该密码算法是入选 2018 年全国密码分组算法设计竞赛第二轮的十个密码算法之一 [27]. 采用广义的 Feistel 结构, 支持 128 比特和 256 比特的明文分组和密钥长度, 密码算法主要包含三个版本: FBC128-128、FBC128-256、FBC256-256. 下面以 FBC128-128 版本进行缩减轮的最小活跃 S 盒搜索, 在 FBC 的轮函数中, 存在着下列 4 输入异或的线性操作.

$$a_{i+1} = v_a \oplus (v_a \lll 3) \oplus (v_a \lll 10) \oplus b_i$$
$$d_{i+1} = v_d \oplus (v_d \lll 3) \oplus (v_d \lll 10) \oplus c_i$$

对于异或输入个数大于 2 的情况, 可以运用文献 [11] 中 2-输入异或方法和不引入辅助变量方法, 也可以运用本文所提出的 3-2 输入异或方法. 下面便分别运用这三种方法对 FBC 算法从第 1 轮到第 12 轮进行活跃 S 盒的搜索, 目标函数的约束条件统一采用顺序编码算法 2 进行约束, 同时都添加定界条件编码. 三种不同方法搜索到 12 轮时的搜索总时间、变量个数、句子数量和最小活跃 S 盒如表 1 所示. 其中变量个数和句子数量为搜索第 12 轮 FBC 算法时, 约束活跃 S 盒为 34 所需的构造 SAT 模型变量和句子, 总搜索时间为第 1 轮搜索到第 12 轮所花费的时间, 其他算法的数据表格相似.

从表 1 可以看出, 运用 3-2 输入异或方法与 2-输入异或方法所产生的句子数量相同, 但前者所产生的变量个数小于后者. 不引入辅助变量方法产生的变量个数是最少的, 但其产生的句子数量却是三者中最多的. 对于总搜索时间而言, 运用 3-2 输入异或方法所需的时间比 2-输入异或方法和不引入辅助变量方法都要少. 应用 SAT 搜索技术, 给出 FBC 算法前 12 轮的最小活跃 S 盒数量, 如表 2 所示. 据我们所知, 尚未有公开的论文给出 FBC 算法前 12 轮最小活跃 S 盒的个数.

[1] https://gitee.com/yghreason/2024pythondraw

表 1 不同异或方法对 12 轮 FBC 搜索对比
Table 1 Comparison of 12 rounds of FBC search using different XOR methods

异或方法	变量个数	句子数量	总搜索时间 (s)	最小活跃 S 盒
2-输入异或方法 [11]	10 093	32 023	3424	34
不引入辅助变量方法 [11]	8557	35 095	3363	34
3-2 输入异或方法	9325	32 023	3189	34

表 2 缩减轮 FBC 算法的最小活跃 S 盒
Table 2 Minimum active S-box of reduced-round FBC algorithm

轮数	1	2	3	4	5	6	7	8	9	10	11	12
最小活跃 S 盒	0	1	2	6	9	13	16	20	23	27	30	34

5.2 不同顺序编码方法的对比

为验证第 3 节提出的顺序编码方法 2 和顺序编码方法 3 的可行性, 分别将两种新的编码方法和文献 [24] 中的顺序编码方法应用于不同密码算法的最小活跃 S 盒搜索, 对比三者间产生的变量、句子数量、搜索花费总时间和最小活跃 S 盒数量. 除了对目标函数约束的编码方法不同外, 其他构造 SAT 模型方法都相同, 不添加定界条件, 保证实验的单一变量原则.

5.2.1 SMS4 分组密码算法搜索

SMS4 算法是国内第一个商用分组密码标准, 是中国无线局域网安全标准所推荐使用的密码算法 [28], 采用的密码结构为非平衡 Feistel 结构, 分组长度和密钥长度均为 128 比特, 加解密的轮数为 32 轮. 下面运用三种不同的顺序编码方法对缩减轮的 SMS4 算法进行 16 轮最小活跃 S 盒的搜索.

从表 3 可以看出, 运用不同顺序编码方法约束目标函数, 所需的变量个数和产生的句子数量不同, 运用顺序编码方法 2 对第 16 轮 SMS4 搜索, 约束活跃 S 盒为 15 时产生的变量个数比顺序编码方法减少了 $\frac{(15-1)\cdot 15}{2} = 105$, 顺序编码方法 3 比顺序编码方法减少了 210 个变量, 相应的顺序编码方法产生的句子数量也多于两种新的顺序编码方法. 在搜索时间方面, 运用顺序编码方法 2 比顺序编码方法搜索时间缩减了 250 s, 顺序编码方法 3 也比顺序编码方法缩减了 559 s. SMS4 算法前 16 轮的最小活跃 S 盒数量, 如表 4 所示, 搜索得到的结果与文献 [29] 中的结果一致.

表 3 不同顺序编码对 16 轮 SMS4 算法搜索对比
Table 3 Comparison of 16 rounds of SMS4 algorithm search with different sequential encoding

约束方法	变量个数	句子数量	总搜索时间 (s)	最小活跃 S 盒
顺序编码方法 [24]	3793	546 579	5105	15
顺序编码方法 2	3688	546 355	4855	15
顺序编码方法 3	3583	546 143	4546	15

表 4 缩减轮 SMS4 算法的最小活跃 S 盒
Table 4 Minimum active S-box of reduced-round SMS4 algorithm

方案	轮数															
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
文献 [29]	0	0	0	1	2	2	5	6	7	8	9	10	10	10	12	15
SMS4 算法	0	0	0	1	2	2	5	6	7	8	9	10	10	10	12	15

5.2.2 PRESENT 分组密码算法搜索

PRESENT 密码是 2007 年由 Bogdanov 等人所提出的一个轻量级 SPN 结构密码算法 [30], 其分块大小为 64 比特, 轮函数迭代次数为 31 轮. 下面运用三种不同的顺序编码方法对 PRESENT 密码算法进行 31 轮搜索最小活跃 S 盒, 对比三者所产生的变量、句子数量和总搜索时间.

从表 5 中可以发现, 当搜索活跃 S 盒达到 62 时, 运用本文的顺序编码方法 3 比顺序编码方法产生的句子减少了将近一万, 相应地, 运用顺序编码方法 3 所花费的总搜索时间也小于其他顺序编码方法. 对于全轮 PRESENT 分组密码的最小活跃 S 盒, 搜索结果如表 6 所示.

表 **5** 不同顺序编码对 31 轮 PRESENT 算法搜索对比

Table 5 Comparison of 31 rounds of PRESENT algorithm search with different sequential encoding

约束方法	变量个数	句子数量	总搜索时间 (s)	最小活跃 S 盒
顺序编码方法 [24]	33 234	83 142	894	62
顺序编码方法 2	31 343	79 299	834	62
顺序编码方法 3	29 452	74 771	823	62

表 **6** PRESENT 算法各轮最小活跃 S 盒

Table 6 PRESENT algorithm's minimum active S-box for each round

方案	轮数															
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
文献 [11]	1	2	4	6	10	12	14	16	18	20	22	24	26	28	30	32
PRESENT 算法	1	2	4	6	10	12	14	16	18	20	22	24	26	28	30	32
方案	**轮数**															
	17	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	
文献 [11]	34	36	38	40	42	44	46	48	50	52	54	56	58	60	62	
PRESENT 算法	34	36	38	40	42	44	46	48	50	52	54	56	58	60	62	

5.2.3 FBC 分组密码算法搜索

将这三种顺序编码方法应用于对 FBC 算法进行 12 轮最小活跃 S 盒的搜索, 比较它们所产生的变量、句子数量和总搜索时间, 实验数据如表 7 所示.

从表 7 可以看出, 顺序编码方法产生的句子数量也多于新的两种顺序编码方法, 在搜索时间方面比其他两者时间要长. 运用顺序编码方法 2 的总搜索时间比顺序编码方法缩减了 1121 s, 而顺序编码方法 3 的搜索效率更高, 比顺序编码方法的总搜索时间减少了 2708 s.

表 **7** 不同顺序编码对 12 轮 FBC 算法搜索对比

Table 7 Comparison of 12 rounds of FBC algorithm search with different sequential encoding

约束方法	变量个数	句子数量	总搜索时间 (s)	最小活跃 S 盒
顺序编码方法 [24]	9886	38 616	14 918	34
顺序编码方法 2	9325	31 191	13 797	34
顺序编码方法 3	8764	30 067	12 210	34

5.3 综合对比

下面接着对不同密码算法运用多种不同方法进行综合搜索对比, 对于 Feistel 结构的密码算法选择 SMS4 作为实验对象, SPN 结构的选择 PRESENT 算法作为实验对象. 第一组实验对于目标函数的约束条件采用顺序编码方法, 添加文献 [11] 中的定界条件编码方法. 第二组实验对目标函数的约束条件采用顺序编码方法 3, 采用本文提出的定界条件编码方法.

5.3.1 SMS4 搜索对比

下面将不同的顺序编码方法和边界编码方法运用于 16 轮的 SMS4 最小活跃 S 盒搜索当中, 实验数据记录于表 8 中.

表 **8** 对 16 轮 SMS4 算法的综合搜索对比

Table 8 Comparison of comprehensive search for 16 rounds of SMS4 algorithm

实验组	变量个数	句子数量	总搜索时间 (s)	最小活跃 S 盒
第一组 [11]	3793	546 985	4458	15
第二组	3583	546 422	4239	15

从表 3 和表 8 的实验数据可以发现, 在添加定界条件编码之后, SAT 模型的句子数量虽然增加了, 但是搜索效率相比于没有添加定界条件编码要快. 从表 8 的实验数据可以看出, 第二组实验产生的句子数量比第一组实验要少, 总搜索时间也比第一组花费时间要少.

5.3.2 PRESENT 搜索对比

接着将不同的顺序编码方法和边界编码方法运用于 31 轮的 PRESENT 算法搜索当中, 实验数据如表 9, 通过与表 5 对比, 可以明显看出在添加定界条件编码之后, 搜索效率显著提高.

表 **9** 对 31 轮 PRESENT 算法的综合搜索对比

Table 9 Comparison of comprehensive search for 31 rounds of PRESENT algorithm

实验组	变量个数	句子数量	总搜索时间 (s)	最小活跃 S 盒
第一组 [11]	33 234	90 073	21.5	62
第二组	29 452	81 599	17.1	62

从表 9 的实验数据可以发现, 在搜索到第 31 轮最小活跃 S 盒为 62 时, 第一组实验所需的总搜索时间仍旧比第二组实验要长.

6 总结

本文回顾了构造 SAT 模型的常用方法, 从多个方面对 SAT 模型进行了优化, 提出新的 k 输入异或模型, 减少构造 SAT 模型所需的变量个数. 为了减少目标函数转换为 SAT 模型所需的变量和句子, 提出了顺序编码方法 2 和顺序编码方法 3, 并改进了定界条件编码方法, 将定界条件约束添加进 SAT 模型提高搜索效率. 然后将新的构造 SAT 模型方法运用于多个密码算法的最小活跃 S 盒搜索中, 并给出多个缩减轮密码算法的活跃 S 盒下界. 虽然本文只将新的构造 SAT 模型方法运用于搜索最小活跃 S 盒, 但同样可以运用于最大差分概率特征搜索和线性搜索等方面. 希望本文提出的构造 SAT 模型有助于自动化搜索的发展和分组密码的设计.

参考文献

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

[2] MATSUI M. Linear cryptanalysis method for DES cipher[C]. In: Advances in Cryptology—EUROCRYPT '93. Springer Berlin Heidelberg, 1994: 386–397. [DOI: 10.1007/3-540-48285-7_33]

[3] MOUHA N, WANG Q J, GU D W, et al. Differential and linear cryptanalysis using mixed-integer linear programming[C]. In: Information Security and Cryptology—INSCRYPT 2011. Springer Berlin Heidelberg, 2012: 57–76. [DOI: 10.1007/978-3-642-34704-7_5]

[4] SUN S W, HU L, WANG P, et al. Automatic security evaluation and (related-key) differential characteristic search: Application to SIMON, PRESENT, LBlock, DES (L) and other bit-oriented block ciphers[C]. In: Advances in Cryptology—ASIACRYPT 2014, Part I. Springer Berlin Heidelberg, 2014: 158–178. [DOI: 10.1007/978-3-662-45611-8_9]

[5] SASAKI Y, TODO Y. New impossible differential search tool from design and cryptanalysis aspects: Revealing structural properties of several ciphers[C]. In: Advances in Cryptology—EUROCRYPT 2017, Part III. Springer Cham, 2017: 185–215. [DOI: 10.1007/978-3-319-56617-7_7]
[6] XIANG Z J, ZHANG W T, BAO Z Z, et al. Applying MILP method to searching integral distinguishers based on division property for 6 lightweight block ciphers[C]. In: Advances in Cryptology—ASIACRYPT 2016, Part III. Springer Berlin Heidelberg, 2016: 648–678. [DOI: 10.1007/978-3-662-53887-6_24]
[7] TODO Y, ISOBE T, HAO Y L, et al. Cube attacks on non-blackbox polynomials based on division property[J]. IEEE Transactions on Computers, 2018, 67(12): 1720–1736. [DOI: 10.1109/TC.2018.2835480]
[8] SUN L, WANG W, WANG M Q. Automatic search of bit-based division property for ARX ciphers and word-based division property[C]. In: Advances in Cryptology—ASIACRYPT 2017, Part I. Springer Cham, 2017: 128–157. [DOI: 10.1007/978-3-319-70694-8_5]
[9] MOUHA N, PRENEEL B. Towards finding optimal differential characteristics for ARX: Application to Salsa20[J]. IACR Cryptology ePrint Archive, 2013: 2013/328. https://eprint.iacr.org/2013/328.pdf
[10] KÖLBL S, LEANDER G, TIESSEN T. Observations on the SIMON block cipher family[C]. In: Advances in Cryptology—CRYPTO 2015, Part I. Springer Berlin Heidelberg, 2015: 161–185. [DOI: 10.1007/978-3-662-47989-6_8]
[11] SUN L, WANG W, WANG M Q. Accelerating the search of differential and linear characteristics with the SAT method[J]. IACR Transactions on Symmetric Cryptology, 2021: 269–315. [DOI: 10.46586/tosc.v2021.i1.269-315]
[12] SUN S W, GERAULT D, LAFOURCADE P, et al. Analysis of AES, SKINNY, and others with constraint programming[J]. IACR Transactions on Symmetric Cryptology, 2017, 2017(1): 281–306. [DOI: 10.13154/tosc.v2017.i1.281-306]
[13] GÉRAULT D, LAFOURCADE P, MINIER M, et al. Computing AES related-key differential characteristics with constraint programming[J]. Artificial Intelligence, 2020, 278: 103183. [DOI: 10.1016/j.artint.2019.103183]
[14] SUN S W, HU L, WANG M Q, et al. Towards finding the best characteristics of some bit-oriented block ciphers and automatic enumeration of (related-key) differential and linear characteristics with predefined properties[J]. IACR Cryptology ePrint Archive, 2014: 2014/747. https://eprint.iacr.org/2014/747.pdf
[15] SASAKI Y, TODO Y. New algorithm for modeling S-box in MILP based differential and division trail search[C]. In: Innovative Security Solutions for Information Technology and Communications—SecITC 2017. Springer Cham, 2017: 150–165. [DOI: 10.1007/978-3-319-69284-5_11]
[16] ZHANG Y J, SUN S W, CAI J H, et al. Speeding up MILP aided differential characteristic search with Matsui's strategy[C]. In: Information Security—ISC 2018. Springer Cham, 2018: 101–115. [DOI: 10.1007/978-3-319-99136-8_6]
[17] MATSUI M. On correlation between the order of S-boxes and the strength of DES[C]. In: Advances in Cryptology—EUROCRYPT '94. Springer Berlin Heidelberg, 1995: 366–375. [DOI: 10.1007/BFb0053451]
[18] RUSSELL S J, NORVIG P. Artificial Intelligence: A Modern Approach[M]. Pearson, 2016. [DOI: 10.5860/choice.33-1577]
[19] SOBOLEV S K. Conjunctive normal form[R/OL]. Encyclopedia of Mathematics, 2020. http://encyclopediaofmath.org/index.php
[20] COOK S A. The complexity of theorem-proving procedures[C]. In: Proceedings of the Third Annual ACM Symposium on Theory of Computing (STOC '71). ACM, 1971: 151–158. [DOI: 10.1145/800157.805047]
[21] ZHANG Z Y, HOU C G, LIU M C. Collision attacks on round-reduced SHA-3 using conditional internal differentials[C]. In: Advances in Cryptology—EUROCRYPT 2023, Part IV. Springer Cham, 2023: 220–251. [DOI: 10.1007/978-3-031-30634-1_8]
[22] SUN L , WANG W, WANG M Q. More accurate differential properties of LED64 and Midori64[J]. IACR Transactions on Symmetric Cryptology, 2018, 2018(3): 93–123. [DOI: 10.13154/tosc.v2018.i3.93-123]
[23] LIU Y W, WANG Q J, RIJMEN V. Automatic search of linear trails in ARX with applications to SPECK and Chaskey[C]. In: Applied Cryptography and Network Security—ACNS 2016. Springer Cham, 2016: 485–499. [DOI: 10.1007/978-3-319-39555-5_26]
[24] SINZ C. Towards an optimal CNF encoding of Boolean cardinality constraints[C]. In: Principles and Practice of Constraint Programming—CP 2005. Springer Berlin Heidelberg, 2005: 827–831. [DOI: 10.1007/11564751_73]
[25] BIERE A, KEPLER J. CaDiCaL at the SAT Race 2019[R/OL]. SAT RACE 2019, 2019. https://fmv.jku.at/papers/Biere-SAT-Race-2019-solvers.pdf
[26] FENG X T, ZENG X Y, ZHANG F, et al. On the lightweight block cipher FBC[J]. Journal of Cryptologic Research, 2019, 6(6): 768–785. [DOI: 10.13868/j.cnki.jcr.000340]
冯秀涛, 曾祥勇, 张凡, 等. 轻量级分组密码算法 FBC[J]. 密码学报, 2019, 6(6): 768–785. [DOI: 10.13868/j.cnki.jcr.

000340]
[27] WU W L. Preface of special issue on block cipher[J]. Journal of Cryptologic Research, 2019, 6(6): 687–689. [DOI: 10.13868/j.cnki.jcr.000333]
吴文玲. 分组密码专刊序言 (中英文)[J]. Journal of Cryptologic Research, 2019, 6(6): 687–689. [DOI: 10.13868/j.cnki.jcr.000333]
[28] DIFFIE W, LEDIN G. SMS4 encryption algorithm for wireless networks[J]. IACR Cryptology ePrint Archive, 2008: 2008/329. https://eprint.iacr.org/2008/329.pdf
[29] LI L C, WU W L, ZHANG L, et al. New method to describe the differential distribution table for large S-boxes in MILP and its application[J]. IET Information Security, 2019, 13(5): 479–485. [DOI: 10.1049/iet-ifs.2018.5284]
[30] BOGDANOV A, KNUDSEN L R, LEANDER G, et al. PRESENT: An ultra-lightweight block cipher[C]. In: Cryptographic Hardware and Embedded Systems—CHES 2007. Springer Berlin Heidelberg, 2007: 450–466. [DOI: 10.1007/978-3-540-74735-2_31]

附录

表 **10** 不同异或方法对 FBC 搜索时间对比
Table 10 Comparison of FBC search time using different XOR methods

轮数	异或方法 (s)		
	2-输入异或方法 [11]	**不引入辅助变量方法** [11]	**3-2 输入异或方法**
1	0.2	0.2	0.2
2	0.2	0.2	0.2
3	0.2	0.2	0.2
4	0.8	0.8	0.8
5	1.6	1.4	1.3
6	4.5	4.6	4.3
7	6.8	5.7	6.1
8	40.2	48.0	45.6
9	58.0	46.2	51.0
10	359.1	407.3	339.5
11	582.9	732.8	508.7
12	2379.2	2116.5	2231.8
全轮	3424.6	3363.8	3189.2

表 **11** 不同顺序编码方法对 SM4 搜索时间对比
Table 11 Comparison of SMS4 search time with different sequential encoding

轮数	顺序编码方法 (s)			轮数	顺序编码方法 (s)		
	顺序编码方法 [11]	**顺序编码方法 2**	**顺序编码方法 3**		**顺序编码方法** [11]	**顺序编码方法 2**	**顺序编码方法 3**
1	0.3	0.2	0.3	9	82.9	59.4	51.4
2	0.5	0.6	0.4	10	91.8	89.8	70.6
3	0.6	0.6	0.6	11	111.7	191.7	126.1
4	0.9	0.9	0.9	12	234.4	221.8	243.5
5	2.0	2.0	1.1	13	9.4	67.4	51.3
6	1.2	1.2	2.6	14	70.4	21.0	35.7
7	11.0	11.2	10.6	15	693.9	521.2	567.3
8	16.8	17.0	18.9	16	3777.5	3658.3	3361.3
全轮	5105.3	4855.3	4546.6				

表 **12** 不同顺序编码方法对 PRESENT 搜索时间对比

Table 12 Comparison of PRESENT search time with different sequential encoding

轮数	顺序编码方法 (s)			轮数	顺序编码方法 (s)		
	顺序编码方法 [11]	顺序编码方法 2	顺序编码方法 3		顺序编码方法 [11]	顺序编码方法 2	顺序编码方法 3
1	0.2	0.2	0.2	17	17.6	17.6	17.5
2	0.2	0.2	0.2	18	18.8	23.0	18.7
3	0.2	0.2	0.2	19	21.2	19.9	23.7
4	0.4	0.4	0.4	20	27.6	23.0	23.0
5	1.1	0.9	0.9	21	28.5	34.3	28.3
6	0.6	0.7	0.6	22	46.0	28.4	31.6
7	0.9	1.0	1.0	23	51.5	53.0	31.8
8	1.3	1.1	1.2	24	53.2	28.1	45.5
9	2.0	2.2	2.0	25	58.2	67.8	50.4
10	2.7	3.0	2.5	26	65.9	82.2	60.0
11	5.1	5.1	4.6	27	82.3	84.9	76.1
12	6.3	5.6	8.2	28	80.5	69.0	80.2
13	8.2	8.2	7.5	29	85.4	75.6	85.5
14	11.4	12.3	12.1	30	93.1	91.3	91.4
15	11.2	10.7	14.5	31	96.1	68.8	89.5
16	16.4	14.2	12.8	全轮	893.8	834.1	823.4

表 **13** 不同顺序编码方法对 FBC 搜索时间对比

Table 13 Comparison of FBC search time with different sequential encoding

轮数	顺序编码方法 (s)			轮数	顺序编码方法 (s)		
	顺序编码方法 [11]	顺序编码方法 2	顺序编码方法 3		顺序编码方法 [11]	顺序编码方法 2	顺序编码方法 3
1	0.2	0.2	0.2	7	22.4	14.2	13.7
2	0.2	0.2	0.2	8	40.2	125.1	144.9
3	0.2	0.2	0.2	9	232.1	343.6	315.3
4	0.8	0.8	0.8	10	1823.0	1610.9	1719.9
5	1.6	1.5	1.5	11	2766.8	1920.2	2243.5
6	7.9	5.9	6.2	12	9676.0	9774.3	7763.7
全轮	14 918.5	13 797.4	12 210.1				

表 **14** 对 16 轮 SMS4 算法的综合搜索时间对比

Table 14 Comparison of comprehensive search time for 16 rounds of SMS4 algorithm

轮数	实验组 (s)		轮数	实验组 (s)	
	第一组 [11]	第二组		第一组 [11]	第二组
1	0.3	0.3	9	41.2	46.2
2	0.5	0.6	10	122.1	172.4
3	0.6	0.6	11	333.0	281.5
4	0.8	0.9	12	358.8	226.4
5	2.0	2.0	13	35.2	11.7
6	3.2	2.5	14	27.3	19.1
7	11.6	10.6	15	401.2	265.9
8	16.9	20.9	16	3105.5	3180.3
全轮	4458.2	4239.5			

表 **15** 对 31 轮 PRESENT 算法的综合搜索时间对比

Table 15 Comparison of comprehensive search time for 31 rounds of PRESENT algorithm

轮数	实验组 (s)		轮数	实验组 (s)	
	第一组 [11]	第二组		第一组 [11]	第二组
1	0.2	0.2	17	0.7	0.5
2	0.2	0.2	18	0.6	0.6
3	0.4	0.4	19	0.5	0.6
4	0.4	0.3	20	0.7	0.6
5	1.1	0.8	21	0.7	0.6
6	0.5	0.4	22	0.7	0.5
7	0.6	0.5	23	0.7	0.5
8	0.5	0.5	24	0.8	0.6
9	0.4	0.3	25	0.8	0.6
10	0.4	0.4	26	0.9	0.7
11	0.4	0.5	27	1.1	0.7
12	0.5	0.5	28	1.0	0.8
13	0.6	0.5	29	1.3	0.8
14	0.6	0.4	30	1.3	0.9
15	0.7	0.5	31	1.6	1.1
16	0.6	0.6	全轮	21.5	17.1

作者信息

颜国华 (2000–), 海南海口人, 硕士研究生. 主要研究领域为对称密码算法的设计与分析.
23151214096@stu.xidian.edu.cn

崔笑 (2000–), 山东济南人, 硕士研究生. 主要研究领域为对称密码算法的设计和安全性分析.
cuixiao2000cx@163.com

王保仓 (1979–), 河南郸城人, 博士, 二级教授, 博士生导师, 西安电子科技大学华山学者领军教授, 中国密码学会理事. 主要研究领域为抗量子公钥密码、全同态密码、密态数据计算、密码分析.
bcwang@xidian.edu.cn

张凤荣 (1982–), 河北邯郸人, 教授, 博士生导师. 主要研究方向为密码函数、对称密码设计与分析等.
zhangfengrong@xidian.edu.cn

韦永壮 (1976–), 广西百色人, 教授. 主要研究领域为密码算法的安全性分析.
walker_wyz@guet.edu.cn