软件学报 ISSN 1000-9825, CODEN RUXUEW
[doi: 10.13328/j.cnki.jos.007686] [CSTR: 32375.14.jos.007686]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

DLAKRS: 面向大状态 SIMECK 分组密码的深度学习辅助密钥恢复方案*

杨亚涛[1,2], 冉小玉[1], 范修凤[1], 王彩冰[1]

[1](北京电子科技学院 电子与通信工程系, 北京 100070)

[2](西安电子科技大学 通信工程学院, 陕西 西安 710071)

通信作者: 杨亚涛, E-mail: yy2008@163.com

摘　要: 近年来, 深度学习为分组密码安全性分析提供了新的研究路径. 然而, 在大状态分组密码场景下, 神经区分器输入维度过高以及密钥搜索空间快速膨胀, 严重制约了深度学习辅助攻击的效率与稳定性. 针对上述问题, 提出面向大状态 SIMECK 分组密码的深度学习辅助密钥恢复方案 DLAKRS. 该方案在训练阶段引入基于梯度的动态敏感比特选择策略, 在保证区分性能的前提下有效压缩神经区分器的输入维度; 在密钥恢复阶段, 结合错误密钥响应分布构建基于统计评分引导的搜索方法, 缓解候选密钥筛选过程中的评分波动. 基于上述方案, 针对 SIMECK 系列大状态轻量级分组密码构建多阶段深度学习辅助密钥恢复攻击方案. 实验结果表明, 在输入维度削减 45.8%–62.5% 的情况下, 区分器准确率损失控制在 10% 以内, 且相较于传统多阶段攻击方法显著降低了计算开销. 在 SIMECK48/96 的 16 轮攻击和 SIMECK64/128 的 18 轮攻击中, 所提方法均表现出稳定的密钥恢复性能. 研究结果表明, 该方案为大状态 SIMECK 分组密码的深度学习辅助密钥恢复提供了一种有效途径.

关键词: 深度学习; SIMECK; 大状态分组密码; 密码分析; 密钥恢复

中图法分类号: TP309

中文引用格式: 杨亚涛, 冉小玉, 范修凤, 王彩冰. DLAKRS: 面向大状态SIMECK分组密码的深度学习辅助密钥恢复方案. 软件学报. http://www.jos.org.cn/1000-9825/7686.htm

英文引用格式: Yang YT, Ran XY, Fan XF, Wang CB. DLAKRS: Deep-learning-assisted Key Recovery Method for Large-state SIMECK Block Ciphers. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/7686.htm

DLAKRS: Deep-learning-assisted Key Recovery Method for Large-state SIMECK Block Ciphers

YANG Ya-Tao[1,2], RAN Xiao-Yu[1], FAN Xiu-Feng[1], WANG Cai-Bing[1]

[1](Department of Electronic and Communication Engineering, Beijing Electronic Science & Technology Institute, Beijing 100070, China)

[2](School of Telecommunications Engineering, Xidian University, Xi'an 710071, China)

Abstract: In recent years, deep learning has provided a new research path for block cipher security analysis. However, for large-state block ciphers, the high input dimensionality of neural distinguishers and the rapid expansion of the key search space severely limit the efficiency and stability of deep learning-assisted attacks. To address these issues, this study proposes DLAKRS, a deep learning-assisted key recovery scheme for large-state SIMECK block ciphers. In the training phase, the scheme introduces a gradient-based dynamic sensitive bit selection strategy to effectively compress the input dimension of the neural distinguisher while ensuring distinguishing performance. In the key recovery phase, a statistical score-guided search method is constructed based on the wrong-key response distribution to alleviate score fluctuations in the candidate key screening process. Based on the above scheme, this study constructs a multi-stage deep learning-assisted key recovery attack scheme for the SIMECK series of large-state lightweight block ciphers. The experimental results show that, when the input dimension is reduced by 45.8%–62.5%, the accuracy loss of the distinguisher is controlled within 10%,

* 基金项目: 国家自然科学基金 (62476013); 中央高校基本科研业务费专项资金 (3282025039, 3282024052)
收稿时间: 2026-01-09; 修改时间: 2026-03-04; 采用时间: 2026-04-22; jos 在线出版时间: 2026-07-01

and the computational cost is significantly reduced compared with traditional multi-stage attack methods. In 16-round attacks on SIMECK48/96 and 18-round attacks on SIMECK64/128, the proposed scheme shows stable key recovery performance. These results show that the proposed scheme provides an effective way for deep learning-assisted key recovery of large-state SIMECK block ciphers.
Key words: deep learning; SIMECK; large-state block cipher; cryptanalysis; key recovery

分组密码是现代密码学中最为基础且应用最为广泛的密码算法之一, 最早的基本概念是由 Shannon[1]于 1949 年提出的. 随着分组密码结构与设计理念的不断发展, 分组密码的安全性分析方法也逐步成熟, 形成了差分密码分析[2]、积分密码分析[3]、线性密码分析[4]以及中间相遇攻击[5]等一系列经典分析技术. 其中, 差分密码分析因其适用性强、攻击效果显著, 长期以来在分组密码安全性评估中占据核心地位.

近年来, 人工智能与机器学习技术的快速发展为分组密码分析提供了新的研究思路. 深度学习方法凭借其自动特征提取和非线性建模能力, 被逐渐引入密码分析领域, 用于从大规模密文数据中挖掘潜在的统计特征, 从而辅助传统密码分析方法开展安全性评估. 2019 年, Gohr[6]首次将深度残差神经网络引入差分密码分析, 通过构建神经差分区分器, 显著提升了对简化轮数 SPECK32/64 算法的攻击效率, 验证了深度学习在分组密码分析中的可行性. 此后, 研究人员围绕神经区分器的结构设计、输入表示以及训练策略等方面展开了深入研究. 进一步地, Chen 等人[7]提出教师-学生区分器模型, 揭示了仅有部分密文比特承载关键区分信息, 为后续基于比特敏感度的特征选择方法奠定了基础.

然而, 当研究对象由小状态分组密码扩展至大状态分组密码时, 现有深度学习辅助密码分析方法面临新的瓶颈. 一方面, 大状态密码中密文比特数量的显著增加导致神经区分器输入维度急剧上升, 使模型训练与推理的计算开销明显增大, 并对区分性能的稳定性产生不利影响. 另一方面, 在密钥恢复阶段, 候选密钥空间随状态规模呈指数级增长, 使得密钥搜索过程在效率和成功率方面均受到严重制约. 如何在保证区分能力的前提下有效降低神经区分器输入维度, 并控制大规模密钥搜索的复杂度, 已成为深度学习辅助大状态分组密码分析亟需解决的关键问题. 已有研究尝试通过多阶段密钥恢复框架缓解上述问题, 但在特征维度压缩与密钥搜索稳定性方面仍存在改进空间.

针对上述问题, 本文提出了一种结合动态敏感比特选择与错误密钥反馈统计评分的深度学习辅助密钥恢复方案. 该方案利用神经网络反向传播过程中的梯度信息动态识别关键密文比特, 实现输入维度的有效压缩. 同时, 通过构建统计评分引导的密钥搜索策略, 提高多阶段密钥恢复过程的稳定性与效率. 本文以 SIMECK 系列大状态轻量级分组密码为研究对象, 构建并验证了多阶段深度学习辅助密钥恢复攻击的有效性.

本文贡献如下:

(1) 提出面向大状态 SIMECK 分组密码的深度学习辅助密钥恢复方案 DLAKRS. 该方案由动态敏感比特选择方法与统计评分引导的密钥搜索方法这两个核心模块组成, 通过敏感特征筛选与统计评分搜索相结合, 实现高效密钥恢复.

(2) 设计一种基于梯度的动态敏感比特选择 (dynamic sensitive bit selection, DSBS) 方法. 不同于比特敏感度测试 (bit sensitivity test, BST) 需要对每个候选比特分别进行扰动并多次前向推理评估其贡献度的方法, 该方法通过捕获神经网络反向传播中的梯度信息, 可在一次计算中同时获得所有输入比特的敏感度, 避免了传统 BST 方法逐比特扰动测试的高昂开销. 同时, DSBS 将敏感度评估嵌入区分器训练过程, 训练完成时即可同步获得稳定的敏感比特选择结果, 计算复杂度降低一个数量级以上.

(3) 构建一种统计评分引导的密钥搜索方法. 通过对错误密钥响应分布进行建模, 实现候选密钥的统计评分与自适应筛选. 该方法能够在显著降低特征维度的同时充分挖掘剩余特征中的统计信息, 有效提升密钥搜索阶段的成功率与稳定性.

(4) 将所提 DLAKRS 方案系统性地应用于 SIMECK48/96 和 SIMECK64/128 大状态轻量级分组密码的多阶段密钥恢复攻击. 分别在 16 轮和 18 轮变体上验证了其有效性. 在汉明距离不超过 3 的判定标准下, 最终成功率均达到 100%, 平均攻击时间为 93.12 s 和 75.17 s, 相较于基础的多阶段框架在效率和稳定性方面均有明显提升.

本文第 1 节介绍基于深度学习的分组密码差分密码分析的相关方法和研究现状. 第 2 节介绍本文所需的基础

知识, 包括 SIMECK 算法、中性比特、神经区分器及密钥恢复攻击和多阶段深度学习密钥恢复框架. 第 3 节介绍本文构建的动态敏感比特与错误密钥响应分布 (wrong-key response profile, WKRP) 统计评分引导的深度学习辅助密钥恢复方案. 第 4 节和第 5 节将方法运用在 SIMECK 密码上来验证方案的有效性. 最后总结全文.

1 相关工作

近年来, 随着人工智能技术的突飞猛进, 其与现代密码学的交叉融合日益深化, 正逐步形成一个极具潜力的前沿跨学科领域. 早期两者结合主要是应用在侧信道分析[8]上, 2019 年, Gohr[6]首次提出一种将神经学习与差分密码分析结合的密码分析技术, 并证明了深度学习在密码分析中的有效性, 通过训练深度残差神经网络构建差分区分器, 显著提升了针对简化轮数 SPECK32/64 密码算法的攻击效率. 该研究不仅实现了优于传统差分密码分析的密钥恢复效果, 成功将对 11 轮 SPECK 的攻击复杂度降至约 2^{38}, 还发现神经网络能捕捉到传统分布表无法利用的密文对分布特征. 该工作的实现为深度学习在密码分析领域的应用做出了开创性贡献.

继 Gohr 之后, Jain 等人[9]开发了一种多层感知机神经区分器来分析轻量级密码 PRESENT, 能够以高概率区分 3–6 轮的加密数据和随机数据. 2021 年, Benamira 等人[10]深入分析了神经区分器的工作原理, 指出神经区分器不仅依赖密文对的差分分布, 还与倒数第 2 轮和倒数第 3 轮的差分分布密切相关, 本质上建立了一个近似差分分布表. 2023 年, Chen 等人[11]开发了一种新的神经网络差分区分器模型, 该模型以多密文对 (multiple ciphertext pairs) 替代单密文对作为输入, 以增强 5–7 轮 SPECK32 的神经区分器性能. Bao 等人[12]提出同时中性比特和条件中性比特, 扩展了中性比特的概念以找到更多的中性比特, 并利用前置差分增加攻击轮数, 成功对 13 轮 SPECK32/64 进行了密钥恢复攻击.

在基于深度学习的密码分析发展过程中, Chen 等人[7]发现 Gohr 的攻击依赖实验, 缺乏理论复杂度估计框架, 并提出教师-学生 (teacher-student) 区分器模型. 他们通过实验发现, 仅有部分密文比特 (即敏感比特) 承载了区分随机样本与密文样本的关键特征. 基于此, 他们提出了比特敏感度测试 BST, 通过比较每个比特对预测结果的贡献度, 筛选出高敏感度的子集来训练学生区分器. 这一思路被 Li 等人[13]进一步扩展到密钥维度, 提出密钥比特敏感性测试 (key bit sensitivity test, KBST). KBST 揭示了并非所有子密钥比特都对区分器的输出产生同等影响, 进而指导攻击者将庞大的密钥空间划分为若干个较小的子空间进行搜索. 这些工作从理论上证明了, 通过牺牲微小的区分精度, 可以换取密钥搜索空间的指数级缩减.

当目标从 SPECK32/64 转向 SPECK64、SPECK128 等大状态密码时, 直接应用 Gohr 的攻击框架变得不可行. Gohr 的原始攻击需要猜测最后一轮的所有子密钥比特以进行部分解密, 对于大状态密码, 这意味着需要遍历 2^{48} 甚至 2^{64} 以上的密钥空间, 计算复杂度过高. 2023 年, 陈怡等人[14]提出了一种深度学习辅助的大状态分组密码密钥恢复框架, 并在大状态 SPECK 上进行了测试. 这为在大状态分组密码上实现深度学习辅助密钥恢复攻击创造了新的可能性. 2024 年, Huang 等人[15]提出可并行的多阶段密钥恢复框架, 对 SPECK64/96 和 SPECK96/96 进行了密钥恢复攻击, 减少了时间复杂度. 2025 年, 申焱天等人[16]在该框架的基础上提出了通用的 NDAFL 增强框架, 该框架利用特征定位技术将计算密集的神经区分器重构为高效的查找表, 从而在大幅降低计算复杂度的同时, 成功实现对 SPECK96 和 SPECK128 算法目前最长轮数的实际全过程密钥恢复攻击.

目前来看, 现有研究主要是从多阶段密钥恢复框架设计、并行化搜索以及推理期加速等角度提升深度学习辅助密码分析的效率. 然而, 在大状态分组密码场景下, 神经区分器输入特征维度过高所带来的冗余问题, 以及多阶段密钥搜索过程中评分波动对恢复稳定性的影响, 仍缺乏系统性的解决方案. 针对上述问题, 本文从训练区分器特征选择与密钥恢复时统计稳定性两个层面出发, 提出一种结合动态敏感比特选择与错误密钥响应分布统计评分引导的深度学习辅助密钥恢复方案 DLAKRS, 为大状态分组密码的高效、稳定攻击提供了一种互补性的解决思路.

2 基础知识

本节将介绍深度学习辅助密钥恢复攻击方案的基础知识、表示符号以及相关概念.

2.1 符号说明

符号说明如表 1 所示.

表 1 符号说明

符号	说明
n	字长 (bits)
$2n$	状态大小 (bits)
rk_r	第 r 轮轮密钥
(x_r, y_r)	表示经过 r 轮轮密钥
$\odot$	比特与运算
$\oplus$	比特异或运算
$x \lll s$	向左旋转移位 s 比特
$x \ggg s$	向右旋转移位 s 比特
$\Delta_{[i]}$	表示一个单比特差分, i 为二进制中1所在位置
$\Delta_{[i,j]}$	表示第 i 位和第 j 位为活跃比特的输入差分
$hw(x,y)$	x和 y 的汉明距离
$\{i\sim j\}$	表示连续比特位置集合, 当$i \leqslant j$时, $\{i\sim j\} = \{i, i+1,\ldots,j\}$

2.2 SIMECK 算法

SIMECK 是一系列轻量级分组密码, 由 Yang 等人[17]于 2015 年设计. SIMECK 旨在结合 SIMON 和 SPECK 密码的优良组件, 以在硬件实现中实现更小的面积开销. SIMECK 系列使用费斯特尔 (Feistel) 结构, 支持 32/64、48/96 和 64/128 这 3 种块大小和密钥大小组合. 轮函数 (round function) 使用循环移位、按位与 (AND) 和异或 (XOR) 的组合 (类似 SIMON). SIMECK32/48/64 分别有 32、36 和 44 轮迭代. 每轮的子密钥是通过复用轮函数结构的非线性密钥调度从主密钥生成的. 图 1 为 SIMECK 密码的轮变换. SIMECK48 和 SIMECK64 之所以被称为“大状态”, 是因为相较于 Gohr[6]最初研究的 SPECK32/64, 它们的输入维度和密钥空间呈指数级增长, 直接套用原始神经区分器会导致计算不可行.

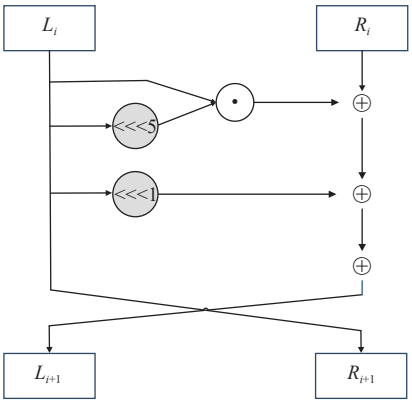

图 1 SIMECK 轮变换图

2.3 中性比特

中性比特是差分密码分析中的重要概念, 最初由 Biham 等人[2]提出. 在差分攻击中, 输入差分为$\Gamma \to \Delta$, 对于满足输入差分 Γ 的明文对 $(P, P\oplus\Gamma)$, 如果同时翻转明文对的第 i 位上的值, 而输出的密文差分仍保持为 Δ 不变, 则这个比特位置 i 被称为中性比特, 第 i 位称为中性位. 设翻转后密文对仍满足给定差分的概率为 p, 当 p 达到某一阈值时, 该比特位置被称为概率性中性比特, p 表示该比特的中性程度.

Bao 等人[12]进一步提出了同时中性比特集 (simultaneous-neutral bit-set, SNBS) 的概念, 即一组包含 d 位的比特位置 $\{i_1, i_2, \ldots, i_d\}$, 当这些位置同时翻转时, 输出差分依然保持不变.

中性比特的价值在于, 给定一个符合输入差分的明文对, 通过对其 k 个中性比特进行组合翻转, 可以快速生成多个同样满足该差分的明文对, 从而显著提升差分攻击的效率.

2.4 神经区分器及密钥恢复攻击

Gohr[6]在 2019 年首次把深度学习与传统差分分析结合, 先选定一个输入差分 ΔP, 在若干轮加密后得到某个目标差分 ΔS, 然后用大量样本训练一个二分类网络 ND_r, 区分两类密文对. 样本为 1 的是具有差分特征正确的密文对, 样本为 0 的是随机密文对. 神经网络学到的其实是真实密码在差分约束下的密文分布与随机置换分布的细微差别, 从而形成一个统计意义上的区分器.

Gohr[6]采用残差网络 (ResNet) 来处理比特向量输入. 输入通常是一个密文对 (或解密 1 轮后的伪密文对) 的比特拼接或差分表示; 输出是一个分数: $Z = ND_r(C_0, C_1) \in [0,1]$, 当 $Z > 0.5$ 时偏向判为正类.

传统差分神经区分器以单个密文对 (C_0, C_1) 为输入输出判别分数 Z. 当密码轮数增大, 分组变大时, 单对样本的统计偏差很弱, 网络输出方差较大, 导致区分能力下降. 多密文对神经区分器把同一输入差分下生成的 m 个密文对组成的 1 个密文结构作为 1 个训练样本, 利用多对累积统计放大非随机特征, 从而提升有效轮数与准确率. Chen 等人[11]提出把多个密文对一起输入网络, 可以提升区分器性能, 尤其在 SPECK 密码上. 主要方法是使用多个密文对组成一个实例作为神经网络的输入. 固定输入差分 ΔP, 一次生成 m 个明文对 $(P_i, P_i \oplus \Delta P)$ 并加密得到密文对集合 $X = \{(C_i^0, C_i^1)\}_{i=1}^m$, 把整个集合视为输入网络, 用网络输出 $Z = f_\theta(X) \in [0,1]$ 来判别该结构更接近真实差分分布还是随机分布.

Gohr[6]的密钥恢复攻击是基于训练好的 r 轮差分神经区分器 ND_r 的. 首先固定输入差分 ΔP, 生成 N 个明文对 $(P_i, P_i \oplus \Delta P)$, 用未知密钥加密得到密文对 (C_i^0, C_i^1). 对每个末轮子密钥猜测 k_g, 对密文对执行 1 轮 (或若干轮) 部分解密, 得到伪密文对 $\left(\widetilde{C_i^0}, \widetilde{C_i^1}\right) = D_{k_g}(C_i^0, C_i^1)$, 并计算区分器输出 $Z_i(k_g) = ND_r\left(\widetilde{C_i^0}, \widetilde{C_i^1}\right) \in [0,1]$. 将所有样本的输出聚合为该猜测的评分, 常用对数似然累加形式:

$$v(k_g) = \sum_{i=1}^{N} \log \frac{Z_i(k_g)}{1 - Z_i(k_g)} \tag{1}$$

对全部 k_g 计算 $v(k_g)$ 并按分数排序, 分数最大的候选对应真实末轮子密钥的概率最高, 随后可在已恢复的末轮子密钥的基础上继续扩展搜索以得到完整密钥. 公式 (1) 为后续引入基于 WKRP 的统计评分提供了自然的统计基础.

2.5 多阶段深度学习密钥恢复框架

由于神经区分器以完整密文对为输入时需要一次性猜测整轮子密钥, 复杂度随分组长度迅速增长, 陈怡等人[14]将区分器和密钥恢复按阶段拆分, 并引入教师-学生区分器和信息比特 (即敏感比特) 概念以缩减每阶段的猜测空间.

● 教师区分器与学生区分器. 对缩减到 r 轮的密码, 使用全部密文比特训练得到的区分器记为教师区分器 ND_r^t, 只取部分密文 (或部分状态) 比特作为输入训练的区分器记为学生区分器 ND_r^s. 学生区分器可视为教师区分器的特例.

● 信息比特 (敏感比特) C_i. 若密文对中第 j 位的变化会显著影响教师区分器的区分准确率, 则该密文位为信息比特, 学生区分器的输入应由信息比特或对应的状态位组成.

● 区分器组合与阶段划分. 设选出 x 个学生区分器构成组合 $\{ND_i\}_{i=1}^x$, 每个区分器对应一个短前置差分:

$$CD_i: \Gamma \to \Delta_i,\ \Pr(CD_i) = p_i \tag{2}$$

并各自关联待恢复的末轮子密钥比特集合 B_i 及其对应的输入比特集合 C_i. 通过分解 (公式 (3)) 将密钥恢复过程拆为 x 个阶段, 每阶段仅恢复 B_i 中的比特.

$$rk = B_1 \cup B_2 \cup \ldots \cup B_x \tag{3}$$

在实际攻击中, 各阶段通常按顺序执行. 第 i 阶段在已知前 $i-1$ 阶段密钥比特的条件下, 仅对 B_i 中的比特进行

枚举猜测, 并利用对应学生区分器 ND_i 对部分解密后的密文进行打分筛选, 从而缩小候选密钥空间. 前一阶段的输出作为先验信息传递至下一阶段, 直至完成全部轮密钥的恢复. 图 2 为多阶段恢复框架图.

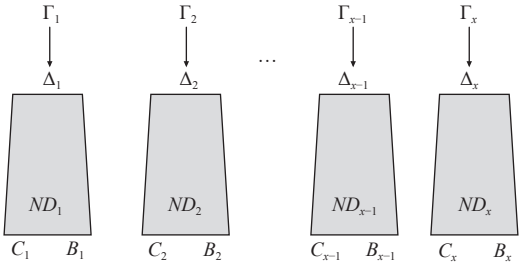

图 2　多阶段恢复框架图

3　DLAKRS 方案设计

3.1　基于梯度显著性分析的动态敏感比特选择方法构建

在大状态分组密码 (如 SIMECK64/128) 的神经区分器训练与推理过程中, 输入维度的急剧增长会导致特征冗余显著增加, 并直接提高模型的计算开销. 传统的比特敏感度测试方法通常通过对每一输入比特进行独立扰动来评估其重要性, 其计算复杂度随输入比特数量增加而线性增长, 在大状态场景下难以高效应用.

为此, 本文提出一种基于神经网络反向传播梯度信息的动态敏感比特选择 (DSBS) 方法, 利用模型内部对输入扰动的响应特性, 对输入比特的重要性进行统一评估, 从而避免逐比特扰动带来的高昂计算代价.

设神经区分器的映射函数如下:

$$z = f(X;\theta) \tag{4}$$

其中, $X \in \{0,1\}^n$ 表示输入密文差分向量, θ 为模型参数. 基于神经区分器输出的判别分数 z 计算其关于输入特征的梯度, 并以梯度绝对值作为比特敏感度度量.

定义第 i 个输入比特的敏感度评分 S_i 为其偏导数绝对值在数据分布 D 上的期望 E:

$$S_i = \mathrm{E}_{X\sim D}\left[\left|\frac{\partial z}{\partial x_i}\right|\right] \tag{5}$$

其中, $\dfrac{\partial z}{\partial x_i}$ 表征了输入空间中第 i 个比特发生微小扰动时对模型判定结果的影响强度. 当第 i 个输入比特发生变化为 Δx_i 时, 根据一阶泰勒展开可得到输出变化近似为 $\Delta z \approx \dfrac{\partial z}{\partial x_i}\Delta x_i$.

因此, 偏导数的绝对值可表示该比特扰动对判别函数输出的局部影响程度, 其数值越大表示该比特对区分真实差分样本与随机样本的贡献越大. 由于输入特征均为 0 或 1, 其梯度绝对值能直接反映各比特对比特输出的敏感度权重.

不同于 BST 所采用的离线逐比特扰动评估方式, DSBS 将敏感度估计过程无缝嵌入神经区分器的训练流程中, 从而在不引入逐比特扰动所需的大量额外前向推理开销下完成特征筛选. 同时从密码结构角度看, 在 SIMECK 密码的 Feistel 结构中, 差分经循环移位、按位与和异或运算后不会均匀扩散, 而是集中在部分状态位上形成更明显的差分偏差. 神经区分器训练收敛后, 这些具有更强判别信息的位置通常对应更大的梯度响应, 因此梯度敏感比特的分布与差分传播特征基本一致.

其具体实现过程如下.

1) 输入密文差分对 X 被展开为取值于 $\{0,1\}$ 的二值向量, 不进行额外归一化处理, 以保持分组密码比特级特征的原始表示形式.

2) 在模型训练的每一个 epoch 中, 利用自动微分机制计算当前 batch 样本对应的梯度向量 $\nabla_X z$, 并对各输入维

度的梯度绝对值进行累积统计. 在反向传播过程中, $\nabla_x z$ 捕捉了差分特征在神经网络复杂权值矩阵中的非线性映射. 高梯度位对应于那些在 Feistel 变换 (尤其是非线性与运算层) 后仍能保持强区分性的位置.

3) 考虑到训练初期梯度分布可能存在随机波动, 本文并非依据单次梯度结果进行比特选择, 而是统计最近 T 个 epoch 中敏感度评分排名前 K 的比特索引集合. 仅当某一比特在多个训练轮次中持续表现出较高敏感度时, 才被最终确认作为高贡献特征比特. 在保持其他实验参数不变的情况下, 对 T 进行简单的敏感性分析. 实验结果证明, 当 T={2~8}时, DSBS 最终筛选得到的敏感比特集合基本保持一致, 说明 T 对最终信息敏感比特选择结果影响较小. 但当 T 过小时, 选择可能容易受到单个 epoch 梯度影响, 导致集合出现不稳定现象. 综合考虑稳定性与计算开销, 本文最终实验中取 $T = 6$ 作为时间窗口长度.

在实际计算中, 对于每一个输入特征维度, 统计其在多个样本上的梯度绝对值平均, 从而得到对应特征的敏感度评分. 由于输入特征由多对密文及其左右分支比特共同构成, 因此需要将特征级敏感度重新聚合到比特级别. 具体地, 对属于同一密文比特的所有特征维度进行归一化求和, 最终得到每一个密文比特的综合敏感度评分.

该过程可形式化表示为:

$$S(b) = \frac{1}{4M} \sum_{i=1}^{M} \sum_{j=1}^{4} \left| \frac{\partial f}{\partial x_{i,b,j}} \right| \tag{6}$$

其中, M 表示密文对数量, b 表示密文比特索引, j 对应左右分支及其差分结构, $x_{i,b,j}$ 表示与密文比特 b 相关的第 j 个输入特征.

通过上述方式, 能够在不引入额外假设的情况下, 从神经网络内部直接提取出对判别最具贡献的密文比特集合. 动态敏感比特选择算法如算法 1 所示.

算法 1. 动态敏感比特选择算法.

输入: 训练数据集 D; 候选密文比特集合 C_{bits}; 训练轮数 N_{ep}; 采样规模 M; 目标比特数 K;
输出: 敏感比特集合 S_{final}; 学生区分器 ND_{stu}.

初始化教师区分器 ND_{tea}, 输入维度全密文比特
$TopK\_history \leftarrow []$ //初始化 Top-K 历史记录列表
for $epoch = 1$ to N_{ep} do
    在数据集 D 上训练 ND_{tea} 一轮
    $X_{\text{sample}} \leftarrow RandomSample(D, M)$ //从训练集随机采样 M 个样本
    with *GradientTape*() as *tape* do
        $preds \leftarrow ND_{\text{tea}}(X_{\text{sample}})$ //前向传播获取预测值
        $grads \leftarrow tape.gradient(preds, X_{\text{sample}})$ //反向传播计算梯度
    end with
    $sensitivity \leftarrow Mean(|\, grads \,|, axis = 0)$ //计算各比特位置的平均梯度绝对值
    $bit\_scores \leftarrow AggregateToBitLevel(sensitivity)$ //聚合至比特级敏感度评分
    $TopK\_bits \leftarrow ArgTopK(bit\_scores, K)$ //选取敏感度最高的 K 个比特
    $TopK\_history.append(TopK\_bits)$ //记录当前轮次的 Top-K 比特
end for
$freq \leftarrow CountFrequency(TopK\_history[-6:])$ //统计最近 6 轮中各比特出现频次
$Sfinal \leftarrow SelectTopK(freq, K)$ //选取频次最高的 K 个比特作为最终集合
初始化学生区分器 ND_{stu}, 输入维度为 $|S_{\text{final}}| \times m \times 4$
for $epoch = 1$ to N_{ep} do
    使用 S_{final} 对应的特征训练 ND_{stu}

20. **end for**
21. **return** S_{final}, ND_{stu} //输出敏感比特集合和学生区分器

算法 1 分为 3 个阶段执行. 第 1 阶段在教师模型训练过程中, 每个 *epoch* 结束后对随机采样的训练样本执行梯度计算, 其中梯度是相对于神经区分器输出的判别分数计算的, 通过计算模型输出对输入特征的梯度绝对值均值得到特征级敏感度, 再聚合到比特级别, 选取 Top-K 敏感比特并记录其演化过程. 第 2 阶段统计最近若干 *epoch* 中 Top-K 比特的出现频率, 选取频率最高的 K 个比特作为最终敏感比特集合. 第 3 阶段使用收缩后的特征集合训练 Student 模型, 得到最终用于密钥恢复的神经区分器.

通过上述动态敏感比特选择过程, 在不依赖人工经验规则或离线扰动假设的前提下, 实现了对神经区分器输入特征的自适应压缩. 与传统静态比特筛选方法相比, DSBS 能够在模型训练过程中持续刻画输入比特的重要性分布, 使所选特征集合更加贴合神经区分器的实际判别能力. 同时在计算效率方面, BST 需要对每一输入比特执行一次完整的前向推理以评估其敏感度, 而 DSBS 仅需一次反向传播即可同时获得所有输入比特的敏感度信息. 从评估次数角度看, BST 方法需要进行 $O(n)$ 次前向推理, 而 DSBS 方法仅需一次反向传播即可获得全部输入比特的敏感度信息. 更为重要的是, DSBS 所得到的高敏感比特集合不仅显著降低了神经区分器的输入维度, 同时也直接约束了密钥恢复阶段所需猜测的子密钥比特范围, 从而在特征层面与密钥搜索复杂度之间建立了明确的对应关系. 这一性质使得 DSBS 成为连接神经区分器构建与高效密钥恢复搜索的关键中间环节, 并为后续引入基于 WKRP 的统计优化策略提供了必要的低维、可搜索空间基础.

3.2 基于 WKRP 的统计评分引导搜索方法设计

在多阶段密钥恢复攻击中, 即使通过 DSBS 显著压缩了神经区分器的输入维度, 候选子密钥空间仍可能随着阶段推进而快速增长. 若仅依据单一评分指标或固定阈值进行筛选, 容易受到神经区分器输出波动和样本规模变化影响, 从而导致搜索效率下降或正确密钥被过早剔除.

为此, 本文引入一种基于错误密钥响应分布 (WKRP) 的统计评分引导搜索方法. 该方法通过对错误密钥条件下神经区分器评分分布进行建模, 对候选子密钥的评分结果进行标准化与融合, 从而在不改变评分函数形式的前提下, 提高候选密钥阈值筛选的稳定性与搜索效率.

具体而言, 对于训练完成的阶段神经区分器 ND, 在给定子密钥猜测 k 的情况下, 对密文结构执行部分解密, 并从中提取由敏感密文比特集合 C_{stage} 所定义的特征向量 X. 为避免数值不稳定, 神经区分器的输出概率被限制在区间 $[\varepsilon, 1-\varepsilon]$ 内.

基于上述输出, 定义子密钥猜测 k 的对数似然累积评分为:

$$LLR_{\text{sum}}(k) = \sum_{j} \log_2 \frac{Z_j}{1 - Z_j} \tag{7}$$

其中, $Z_j = ND(X_j)$ 表示神经区分器对第 j 个结构样本的输出概率. 该评分反映了在当前子密钥假设下, 观测到的结构样本整体上与真实差分分布的一致程度. 当子密钥猜测正确时, 其对应的对数似然累积值在统计意义上通常显著高于错误密钥情形.

为刻画错误密钥下评分函数的统计行为, 本文引入错误密钥响应分布. 具体做法是: 随机采样大量错误子密钥, 对每个错误密钥计算其对应的对数似然评分, 并统计其分布特性. 相关实验表明, 在错误密钥条件下, 评分 $LLR_{\text{sum}}(k)$ 可近似服从正态分布, 其评分函数为:

$$z(k) = \frac{LLR_{\text{sum}}(k) - \mu}{\sigma} \tag{8}$$

其中, μ 和 σ 分别表示错误密钥评分分布的均值和标准差, 由 WKRP 预处理阶段估计得到. 该指标刻画了当前子密钥评分相对于错误密钥统计分布的偏离程度, 可用于衡量其统计显著性.

综合考虑评分的绝对大小与其相对于错误密钥分布的偏离程度, 本文采用如下加权评分函数 $score(k)$ 对候选子密钥进行筛选:

$$score(k) = \alpha \cdot LLR_{\text{sum}}(k) + (1-\alpha) \cdot z(k) \tag{9}$$

其中, $\alpha \in (0,1)$ 为权重参数, 用于平衡对数似然累积值与其相对于错误密钥响应均值的偏离程度. 在每一阶段的密钥搜索过程中, 仅保留评分超过给定阈值 τ 的候选子密钥进入后续搜索. 为进一步控制搜索空间规模, 可在满足阈值条件的候选集合中选取评分最高的 K 个密钥作为下一阶段的输入.

在多阶段密钥恢复过程中, 本文在 DSBS 所约束的低维子密钥空间内, 利用上述 WKRP 的统计评分函数对候选密钥进行筛选, 并仅保留评分超过给定阈值的少量高潜力候选进入下一阶段搜索. 该策略在显著降低搜索复杂度的同时, 保持了较高的密钥恢复成功率.

3.3 多阶段密钥恢复攻击算法设计

本文将 DSBS 方法与基于 WKRP 的统计优化相结合, 构建完整的多阶段密钥恢复攻击框架. 攻击采用 $1+s+r+1$ 结构, 其中 s 表示前置差分轮数, r 表示神经区分器覆盖的轮数. 攻击流程包括离线准备和在线攻击两个阶段.

离线准备阶段使用 DSBS 方法训练各阶段神经区分器, 并为每个区分器预计算 WKRP; 在线攻击阶段按多个阶段串行执行, 每个阶段生成明文结构, 利用中性比特扩展, 并通过基于 WKRP 的统计评分筛选候选密钥. 算法 2 给出了完整的攻击流程.

算法 2. 多阶段密钥恢复攻击算法.

输入: 攻击阶段数 x; 神经区分器集合 $\{ND_1,\ldots,ND_x\}$; 敏感密文比特集合 $\{C_1,\ldots,C_x\}$; 输入差分集合 $\{\Delta_1,\ldots,\Delta_x\}$; 中性比特集合 $\{N_1,\ldots,N_x\}$; 评分阈值集合 $\{\tau_1,\ldots,\tau_x\}$; 最大结构数 N_{struct}; Top-K 参数 K;
输出: 恢复的子密钥 kg 或 FAIL.

1. $kg \leftarrow \varnothing$ //初始化已恢复密钥为空集
2. **for** $stage = 1$ to x do //遍历每个攻击阶段
3. **for** $iter = 1$ to N_{struct} do //遍历每个明文结构
4. $(P_0, P_1) \leftarrow GeneratePlaintextPair(\Delta_{stage})$ //生成满足输入差分的明文对
5. $Struct \leftarrow ExpandWithNeutralBits(P_0, P_1, N_{stage})$ //利用中性比特扩展明文结构
6. $CTS \leftarrow Encrypt(Struct)$ //加密获得密文结构
7. **if** $iter = 1$ then $(\mu, \sigma) \leftarrow BuildWKRP(ND_{stage}, CTS)$ //首次迭代时构建 WKRP
8. **end if**
9. $candidates \leftarrow []$ //初始化候选密钥列表
10. **for** each $k_{stage} \in K_{stage}$ do //遍历当前阶段的候选子密钥
11. $kg_{\text{full}} \leftarrow Combine(k_{stage}, kg)$ //组合当前猜测与已恢复密钥
12. $D \leftarrow DecryptOneRound(CTS, kg_{\text{full}})$
13. $X \leftarrow ExtractFeatures(D, C_{stage})$ //提取敏感比特特征
14. $Z \leftarrow Clip(ND_{stage}(X), \varepsilon, 1-\varepsilon)$
15. $LLR_{\text{sum}} \leftarrow \sum_j \log_2(Z_j/(1-Z_j))$ //计算对数似然比累积评分
16. //基于 WKRP 的统计评分
17. $z \leftarrow (LLR_{\text{sum}} - \mu)/\sigma$ //计算标准化 z 分数
18. $score \leftarrow \alpha \cdot LLR_{\text{sum}} + (1-\alpha) \cdot z$ //计算加权评分
19. **if** $score > \tau_{stage}$ then $candidates.append((kg_{\text{full}}, score))$ //超过阈值加入候选
20. **end if**
21. **end for**
22. **if** $candidates \neq \varnothing$ then //若存在有效候选

23. $kg \leftarrow SelectTopK(candidates, K)$
24. **break**
25. **end for**
26. //当前阶段未找到有效候选密钥
27. **if** *candidates* = ∅ then return FAIL //攻击失败
28. **end if**
29. **end for**
30. **return** *kg*

算法 2 的核心思想是在每个攻击阶段, 首先生成满足前置差分的明文结构并利用中性比特进行扩展, 随后遍历当前阶段的候选子密钥空间, 对每个猜测执行一轮解密、特征提取和神经区分器评分, 最后基于 WKRP 的统计评分与阈值筛选模式保留高潜力候选密钥, 并将其传递至下一阶段. 相较于传统的完全遍历搜索, 该方法利用错误密钥统计先验对候选密钥进行校准筛选, 有效减少了需要评估的候选数量.

3.4 复杂度分析

设攻击过程被划分为 x 个阶段. 在第 i 个阶段中, 需要恢复 $|B_i|$ 个密钥比特, 并使用对应的神经区分器对候选密钥进行评估. 设该阶段最多使用 N_i 个明文结构, 且上一阶段保留的候选子密钥数量为 β_i.

在该阶段中, 需要评估的子密钥数量为 $\beta_i \cdot 2^{|B_i|}$. 对每一个子密钥猜测, 需要对结构中的样本执行部分解密并调用神经区分器进行推理. 因此, 第 i 阶段的时间复杂度可表示为:

$$T = \mathrm{O}\left(\beta_i \cdot 2^{|B_i|} \cdot N_i \cdot N_s \cdot C_i\right) \tag{10}$$

其中, C_i 表示单次神经区分器推理及相关预处理操作的计算代价, N_s 表示每个结构中参与评分的样本规模.

综合所有阶段, 攻击的总体时间复杂度为:

$$T = \sum_{i=1}^{x} T_i \tag{11}$$

在该方法下, DSBS 通过降低输入维度来减少神经网络推理开销, WKRP 的统计评分方法通过 WKRP 减少候选测试数量, 两者协同提升攻击效率.

4 针对 SIMECK48/96 分组密码的密钥恢复攻击实验

4.1 实验参数与配置

针对 SIMECK48/96 的 13 轮神经区分器训练, 本文借鉴文献 [18] 中的相关参数, 采用以下参数配置: 训练数据量为 10^7 个样本, 验证数据量为 10^6 个样本, 每批次训练样本数为 4000, 总训练轮次为 10 个 epoch, 采用多密文对输入格式, 每个实例包含 $m = 8$ 对密文对.

神经网络结构采用残差卷积网络, 具体包含: 1 层 1×1 卷积层 (32 个滤波器)、3 个残差块 (每块包含 2 层 3×3 卷积)、全连接层 (128 → 64 → 1). 所有卷积层均采用 L2 正则化 $\left(\lambda = 10^{-4}\right)$, 激活函数为 ReLU, 输出层采用 Sigmoid 激活函数. 学习率采用周期调度策略:

$$lr_i = \alpha + (n - i) \bmod (n + 1) / n \times (\beta - \alpha) \tag{12}$$

其中, $\alpha = 10^{-4}$, $\beta = 10^{-3}$, $n = 9$. 该周期学习率策略能够帮助模型跳出局部最优解, 提升最终收敛精度.

为覆盖多阶段密钥恢复攻击的需求, 针对 SIMECK48/96 选择出 3 个不同输入差分的神经区分器: $ND_{10}^{13r}(\Delta_{[10]})$、$ND_{18}^{13r}(\Delta_{[18]})$ 和 $ND_{23}^{13r}(\Delta_{[23]})$, 其输入差分分别对应单比特差分的第 10、18、23 位, 用于后续攻击的不同阶段. 本文在相同的 SIMECK 软硬件环境下, 复现并适配了文献 [14] 的多阶段密钥恢复框架作为基线对照组.

4.2 区分器的 DSBS 分析

表 2 汇总了 3 个神经区分器的 DSBS 实验结果, 并与传统的全密文输入方式进行对比. 图 3 为 SIMECK48/96

的教师区分器与学生区分器性能对比图. 从实验结果可以得出以下结论.

(1) 敏感比特与输入差分的关联性: 3 个区分器的敏感比特集合与其输入差分位置呈明显关联性. $ND_{10}^{13r}(\Delta_{[10]})$ 的敏感比特集中在{0~12}, $ND_{18}^{13r}(\Delta_{[18]})$ 集中在{9~18}, $ND_{23}^{13r}(\Delta_{[23]})$ 集中在{14~23}. 这一现象与 SIMECK 轮函数的扩散特性相符: 循环左移 5 位和 1 位使得差分信息主要向低位方向传播, 因此敏感比特集中在差分位置及其低位邻域. 同时, 轮函数中的与运算会限制差分的有效传播路径数量, 使高概率差分主要分布在差分领域范围内.

(2) 维度压缩效果显著: ND_{10}^{13r} 的输入维度从 768 压缩至 416, 压缩率达到 45.8%, ND_{18}^{13r} 和 ND_{23}^{13r} 从 768 压缩至 320, 压缩率为 58.3%. 这意味着在多阶段密钥恢复攻击中, 每个阶段需要猜测的密钥比特数可以相应减少, 从而大幅降低攻击的时间复杂度.

(3) 准确率 (accuracy) 损失可控: ND_{10}^{13r} 和 ND_{18}^{13r} 的学生区分器模型准确率损失分别为−0.01% 和−0.51%, ND_{23}^{13r} 的损失为 2.67%. 平均而言, DSBS 方法在大幅压缩输入维度的同时, 仅造成约 0.72% 的平均准确率变化, 证明了该方法的有效性.

表 2　SIMECK48 神经区分器 DSBS 实验结果汇总

区分器	输入差分	Top-K	初始维度	DSBS敏感比特	压缩维度	准确率变化
ND_{10}^{13r}	$\Delta_{[10]}$	13	768	{0~12}	416	61.33%→61.34%
ND_{18}^{13r}	$\Delta_{[18]}$	10	768	{9~18}	320	60.20%→60.71%
ND_{23}^{13r}	$\Delta_{[23]}$	10	768	{14~23}	320	63.40%→60.73%

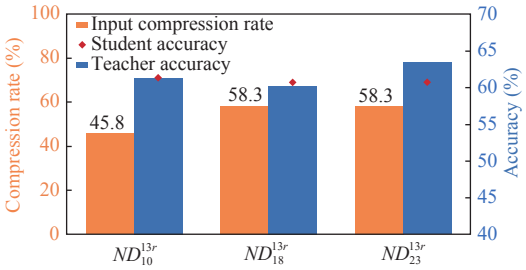

图 3　SIMECK48/96 的教师区分器与学生区分器性能对比图

从表 2 和图 3 可以看出, DSBS 方法能够在将输入维度压缩 45.8%–58.3% 的情况下, 仅造成平均 0.72% 的准确率损失. 更重要的是, DSBS 方法同样揭示了神经区分器所关注的密文比特位置与输入差分之间的内在关联, 为理解深度学习辅助密码分析的工作机理提供了更高效的动态评估方式.

为了验证本文提出的 DSBS 方法的可靠性, 对比了其与传统 BST 方法筛选出的敏感比特集合. 实验发现, 在相同参数 K 下, 两种方法确定的核心敏感比特索引重合度很高. 这证明了 DSBS 在大幅提升评估效率的同时, 能够较好地保留 BST 的特征筛选质量. 因此, 后续密钥恢复攻击阶段的成功率和耗时仅取决于区分器本身的性能, 在 DSBS 与 BST 下的表现具有等效性. 本文后续仅展示 DSBS 优化后的攻击结果.

为验证 DSBS 方法的计算效率优势, 本文将其与传统 BST 方法进行对比. 对比实验环境为: Ubuntu 18.04 操作系统, Intel Xeon Platinum 8358P 处理器, NVIDIA GeForce RTX 3090 GPU (24 GB 显存), TensorFlow 2.5.0 深度学习框架. 表 3 展示了两种方法在 SIMECK48/96 上的计算开销.

表 3　DSBS 与 BST 计算效率对比 (SIMECK48/96)

方法	评估方式	实测时间	备注
BST[7]	逐维度扰动测试	22.58 s	需完整训练后离线执行
DSBS	梯度反向传播	约2 s	训练过程中同步完成

从表 3 可以看出, BST 方法对 96 个密文比特输入进行完整敏感度评估需要约 22.58 s, 而 DSBS 方法仅需约 2 s 即可完成, 加速比超过 10 倍. 这里计算的是敏感度评估阶段的运行时间, 不包括训练数据生成与区分器训练时间. 更重要的是, Chen 等人[7]提出的 BST 以及 Li 等人[13]提出的 KBST 都属于离线敏感度评估方法, 需要在区分器训练完成后对输入比特或密钥比特进行单独执行, 而 DSBS 将敏感度评估嵌入训练过程, 无需额外的评估阶段. 因此, 在保证敏感比特筛选效果的同时, DSBS 能够显著降低敏感度分析阶段的计算开销, 并简化整体攻击流程. 综合考虑计算效率和流程复杂度两个方面, DSBS 方法相较 BST 和 KBST 方法在大状态分组密码分析中具有更好的实用性.

此外, 申焱天等人[16]提出的 NDAFL 框架主要从区分器推理效率角度来优化深度学习辅助密码分析, 而本文工作则侧重于输入特征选择与密钥评分机制, 两者在优化方向上具有一定的互补性.

4.3 SIMECK48 的密钥恢复攻击测试与分析

基于上述区分器训练工作, 本文设计了针对 SIMECK48/96 的 16 轮密钥恢复攻击. 攻击采用 1+1+13+1 结构, 即 1 轮输入差分扩展、1 轮前置差分传播、13 轮神经区分器以及 1 轮免费解密. 攻击任务是恢复最后一轮子密钥的全部比特信息, 划分为 3 个阶段执行. 表 4 列出了完整的攻击组件与参数设置.

表 4　SIMECK48/96 多阶段密钥恢复攻击组件与参数

阶段	区分器	前置差分	中性比特位置	猜测比特数	恢复轮密钥比特
1	ND_{10}^{13r}	$\Delta_{[11,34]}$	{1~10}	11	{0~10}
2	ND_{18}^{13r}	$\Delta_{[19,42]}$	{9~18}	9	{11~19}
3	ND_{23}^{13r}	$\Delta_{[0,47]}$	{14~23}	4	{20~23}

攻击参数设置如下: 每个阶段的前置差分概率均为 2^{-2}; 打分阈值设置为 $C_1 = C_2 = 60$、$C_3 = 40$; 基于 WKRP 统计评分引导搜索迭代次数为 2^5; 密文结构数为 2^5; 每个阶段使用 10 个中性比特扩展明文结构.

在上述配置下, 本文对 SIMECK48/96 实施了 100 次独立的密钥恢复攻击实验. 实验环境为: Ubuntu 18.04 操作系统, Intel Xeon Platinum 8358P 处理器, NVIDIA GeForce RTX 3090 GPU (24 GB 显存), TensorFlow 2.5.0 深度学习框架.

攻击成功的判定标准为猜测密钥 kg 与真实密钥 rk 的汉明距离不超过 3, 即 $hw(kg, rk)\leqslant 3$. 表 5 展示了详细的实验结果统计.

表 5　SIMECK48/96 密钥恢复攻击结果对比

方法	攻击轮数	平均成功率 (%)	平均时间 (s)	实验次数
文献[14]中的多阶段框架	16	96.0	112.29	100
本文中DSBS+WKRP统计评分	16	100.0	93.12	100

从表 5 可以看出, 本文提出的基于 WKRP 的统计优化在攻击成功率以及攻击时间两个维度上优于普通密钥恢复攻击.

(1) 成功率提升: 攻击成功率从 96.0% 提升至 100.0%, 实现了完美的攻击成功率. 这主要归功于 WKRP 统计评分函数能够更稳定地保留高潜力候选密钥.

(2) 攻击时间降低: 平均攻击时间从 112.29 s 降低至 93.12 s, 减少了约 17.1%. 这一提升主要得益于 WKRP 模型提供的统计评分校准机制设置, 使得攻击能够更快地收敛至正确密钥.

表 6 展示了 100 次攻击实验中,猜测密钥与真实密钥之间的汉明距离分布情况.

表 6　攻击结果汉明距离分布

指标	$hw(kg, rk)=0$	$hw(kg, rk)=1$	$hw(kg, rk)=2$	$hw(kg, rk)=3$
实验次数	41	48	10	1
占比 (%)	41	48	10	1

从汉明距离分布可以看出, 在 100 次独立实验中, 41% 的实验能够完全恢复正确密钥 ($hw(kg, rk)=0$), 89% 的实验猜测误差不超过 1 位 ($hw(kg, rk)\leqslant 1$), 且所有实验的汉明距离均不超过 3, 满足攻击成功的判定标准. 上述结果表明, 所提方法在给定攻击模型和判定标准下, 能够稳定地将密钥搜索空间显著压缩.

5 针对 SIMECK64/128 分组密码的密钥恢复攻击实验

5.1 神经区分器训练与 DSBS 分析

本节针对 SIMECK64/128 训练 15 轮神经区分器, 借鉴文献 [18] 中的相关参数, 训练参数与 SIMECK48/96 相同. 选取 3 种单比特差分 $\Delta_{[11]}$、$\Delta_{[22]}$ 和 $\Delta_{[31]}$, 分别对应 3 个攻击阶段. 表 7 展示了 DSBS 分析后各阶段区分器的配置.

表 7　SIMECK64/128 阶段神经区分器配置

阶段	区分器	输入差分	DSBS敏感比特	输入维度	验证准确率
1	ND_{11}^{15r}	$\Delta_{[11]}=(0,0x00000800)$	{0~11}	384	0.726
2	ND_{22}^{15r}	$\Delta_{[22]}=(0,0x00400000)$	{10~22}	416	0.736
3	ND_{31}^{15r}	$\Delta_{[31]}=(0,0x80000000)$	{19~31}	416	0.737

值得注意的是, SIMECK64/128 的 15 轮神经区分器准确率达到 0.727–0.737, 明显高于 SIMECK48/96 的 13 轮区分器. 这一现象可能源于以下原因: SIMECK64 具有更大的状态空间, 使差分特征的传播模式更为丰富, 从而让神经网络能够捕获更多的判别信息. 从算法结构角度看, 状态字长由 24 扩展至 32 时, 参与轮函数非线性运算的比特数量增加, 差分传播路径更加多样, 因此更容易形成可区分的统计特征, 使神经区分器更容易捕获判别特征. 同样图 4 为 SIMECK64/128 的教师区分器与学生区分器性能对比图.

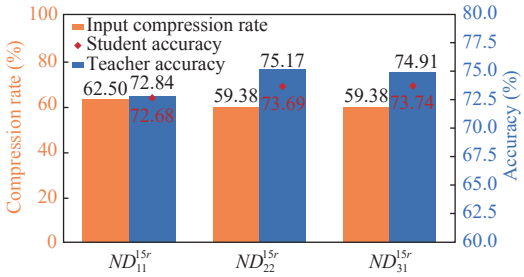

图 4　SIMECK64/128 的教师区分器与学生区分器性能对比图

从图 4 可以看出, SIMECK64/128 的 DSBS 实验呈现以下特点.

(1) 维度压缩效果显著: 3 个区分器的输入维度从 1 024 分别压缩至 384 和 416, 压缩率达到 59.4%–62.5%, 与 SIMECK48 的压缩效果相当.

(2) 准确率损失可控: ND_{11}^{15r} 的准确率损失仅为 0.16%, ND_{22}^{15r} 和 ND_{31}^{15r} 的损失分别为 1.48% 和 1.17%, 平均损失约 0.94%. 这一损失水平与 SIMECK48 实验基本一致, 表明 DSBS 方法在不同分组长度的密码上具有稳定的性能表现.

5.2 多阶段密钥恢复攻击测试与分析

基于上述信息, 设计了 SIMECK64/128 的 18 轮密钥恢复攻击, 采用 1+1+15+1 结构. 攻击划分为 3 个阶段, 每个阶段恢复约 10–12 比特的密钥信息. 表 8 列出了完整的攻击组件与参数设置.

攻击参数设置如下: 每个阶段的前置差分概率均为 2^{-2}; 打分阈值统一设置为 $C_1=C_2=C_3=60$; WKRP 统计评分迭代次数为 2^5; 密文结构数为 2^5; 每个阶段使用 9 个中性比特扩展明文结构. 由于神经区分器准确率较高, 打分阈值相应提高以过滤更多的错误密钥.

本文在 NVIDIA GeForce RTX 3090 GPU 上进行了 100 次独立攻击实验. 攻击成功的判定标准为: 猜测的轮密钥与真实轮密钥的汉明距离不超过 3. 表 9 展示了 SIMECK64/128 攻击的实验结果.

表 8 SIMECK64/128 的 3 阶段密钥恢复攻击配置

阶段	区分器	前置差分	中性比特位置	猜测比特数	恢复轮密钥比特
1	ND_{11}^{15r}	$\Delta_{[12,11]}$	{1~9}	12	{0~11}
2	ND_{22}^{15r}	$\Delta_{[22,23]}$	{12~20}	10	{12~21}
3	ND_{31}^{15r}	$\Delta_{[31,0]}$	{21~29}	10	{22~31}

表 9 SIMECK64/128 密钥恢复攻击实验结果

方法	攻击轮数	攻击结构	成功率 (%)	平均时间 (s)	实验次数
文献[14]中多阶段框架	18	1+1+15+1	98.0	109.45	100
本文DSBS+WKRP统计评分	18	1+1+15+1	100.0	75.17	100

实验结果表明, 本文提出的 DSBS 与 WKRP 统计评分集成方法在 SIMECK64/128 这一更大状态的实例上表现出优异的鲁棒性. 在 100 次独立攻击实验中, 均在工程判定标准 ($hw(kg, rk)\leqslant 3$) 下实现 100% 成功, 且平均攻击时间仅为 75.17 s. 相较于 SIMECK48/96, 该变体虽具有更大的密钥空间, 但由于 DSBS 有效压缩了特征维度且 WKRP 提供了更精准的评分过滤以及中性比特数量不同, 攻击效率并未显著下降. 这一结果有力地验证了本文方法在处理不同分组长度及复杂密钥搜索空间时的通用性与有效性.

表 10 展示了 100 次攻击实验中, 猜测密钥与真实密钥之间汉明距离的分布情况.

表 10 SIMECK64/128 攻击结果汉明距离分布

指标	$hw(kg, rk)=0$	$hw(kg, rk)=1$	$hw(kg, rk)=2$	$hw(kg, rk)=3$
实验次数	35	43	22	0
占比 (%)	35	43	22	0

从汉明距离分布可以看出: 35% 的实验能够完全恢复正确密钥; 78% 的实验猜测误差不超过 1 位; 全部 100 次实验的汉明距离均不超过 2, 攻击精度优异. 值得注意的是, SIMECK64/128 由于字长更大, 单阶段需要猜测的密钥比特数更多, 但本文方案仍实现了 100% 的攻击成功率, 体现了 WKRP 的统计评分函数和 DSBS 策略在大密钥空间场景下的鲁棒性.

需要说明的是, 本文方案测试结果中 100% 攻击成功率是在有限轮数与宽松判定标准 ($hw(kg, rk)\leqslant 3$) 下获得的, 主要用于比较不同搜索策略在相同攻击模型下的相对性能提升, 而非声称对完整轮数 SIMECK 的实际安全性构成威胁. 实验结果表明, 在 SIMECK64/128 这一最大状态实例上, 本文方案在工程判定标准 ($hw(kg, rk)\leqslant 3$) 下实现了 100% 的成功率. 虽然完全恢复率 ($hw(kg, rk)=0$) 为 35%, 但根据多阶段密钥恢复理论, 将搜索空间压缩至 ($hw(kg, rk)\leqslant 3$) 意味着剩余密钥空间已缩减至 $2^{18.3}$ 左右, 攻击者可利用已生成的正确密文结构, 配合短轮神经区分器以极低的计算代价完成最终的比特修正, 这种策略在处理大密钥空间时比暴力搜索更加有效.

6 总 结

针对深度学习辅助密钥恢复攻击在大状态轻量级分组密码中存在的输入维度高和密钥搜索效率不足等问题, 本文提出了一种结合动态敏感比特选择与 WKRP 统计评分优化的多阶段密钥恢复攻击方案 DLAKRS, 并将其运用在 SIMECK 大状态密码上. 该方案在不改变原有攻击模型与轮数设置的前提下, 通过特征选择与统计评分机制的协同设计, 有效提升了攻击过程的效率与稳定性. 在 SIMECK48/96 和 SIMECK64/128 实例上的实验结果表明, 所提方案在相同攻击设置下, 相较于基础多阶段密钥恢复框架, 在攻击成功率和平均攻击时间等指标上均表现出一定优势. 未来可在保持分析框架的前提下, 将该方法进一步扩展至其他轻量级分组密码或更复杂的攻击场景, 并探索其在更高轮数条件下的应用效果.

References

[1] Shannon CE. Communication theory of secrecy systems. The Bell System Technical Journal, 1949, 28(4): 656–715. [doi: 10.1002/j.1538-7305.1949.tb00928.x]

[2] Biham E, Shamir A. Differential cryptanalysis of DES-like cryptosystems. Journal of Cryptology, 1991, 4(1): 3–72. [doi: 10.1007/BF00630563]

[3] Knudsen L, Wagner D. Integral cryptanalysis. In: Proc. of the 9th Int'l Workshop on Fast Software Encryption (FSE 2002). Leuven: Springer, 2002. 112–127. [doi: 10.1007/3-540-45661-9_9]

[4] Matsui M. Linear cryptanalysis method for DES cipher. In: Proc. of the 1993 Workshop on the Theory and Application of Cryptographic Techniques (EUROCRYPT 1993). Lofthus: Springer, 1993. 386–397. [doi: 10.1007/3-540-48285-7_33]

[5] Diffie W, Hellman ME. Exhaustive cryptanalysis of the NBS data encryption standard. In: Slayton R, ed. Democratizing Cryptography: The Work of Whitfield Diffie and Martin Hellman. New York: ACM Press, 2022. 391–414. [doi: 10.1145/3549993.3550008]

[6] Gohr A. Improving attacks on round-reduced SPECK32/64 using deep learning. In: Proc. of the 39th Annual Int'l Cryptology Conf. on Advances in Cryptology (CRYPTO 2019). Santa Barbara: Springer, 2019. 150–179. [doi: 10.1007/978-3-030-26951-7_6]

[7] Chen Y, Shen YT, Yu HB. Neural-aided statistical attack for cryptanalysis. The Computer Journal, 2023, 66(10): 2480–2498. [doi: 10.1093/comjnl/bxac099]

[8] Maghrebi H, Portigliatti T, Prouff E. Breaking cryptographic implementations using deep learning techniques. In: Proc. of the 6th Int'l Conf. on Security, Privacy, and Applied Cryptography Engineering (SPACE 2016). Hyderabad: Springer, 2016. 3–26. [doi: 10.1007/978-3-319-49445-6_1]

[9] Jain A, Kohli V, Mishra G. Deep learning based differential distinguisher for lightweight cipher PRESENT. 2020. https://eprint.iacr.org/2020/846.pdf

[10] Benamira A, Gerault D, Peyrin T, Tan QQ. A deeper look at machine learning-based cryptanalysis. In: Proc. of the 40th Annual Int'l Conf. on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2021). Zagreb: Springer, 2021. 805–835. [doi: 10.1007/978-3-030-77870-5_28]

[11] Chen Y, Shen YT, Yu HB, Yuan ST. A new neural distinguisher considering features derived from multiple ciphertext pairs. The Computer Journal, 2023, 66(6): 1419–1433. [doi: 10.1093/comjnl/bxac019]

[12] Bao ZZ, Guo J, Liu MC, Ma L, Tu Y. Enhancing differential-neural cryptanalysis. In: Proc. of the 28th Int'l Conf. on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2022). Taipei: Springer, 2022. 318–347. [doi: 10.1007/978-3-031-22963-3_11]

[13] Li XW, Ren JJ, Chen SZ. Improved deep learning aided key recovery framework: Applications to large-state block ciphers. Frontiers of Information Technology & Electronic Engineering, 2024, 25(10): 1406–1420. [doi: 10.1631/FITEE.2300848]

[14] Chen Y, Bao ZZ, Shen YT, Yu HB. A deep learning-aided key recovery framework for large-state block ciphers. Scientia Sinica Informationis, 2023, 53(7): 1348–1367 (in Chinese with English abstract). [doi: 10.1360/SSI-2022-0298]

[15] Huang TR, Li YY, Fu QG, Chen YC, Song L. Improving differential-neural cryptanalysis for large-state SPECK. In: Proc. of the 26th Int'l Conf. on Information and Communications Security (ICICS 2024). Mytilene: Springer, 2024. 40–57. [doi: 10.1007/978-981-97-8798-2_3]

[16] Shen YT, Chen Y, Yu HB. General enhancing framework for deep learning-aided cryptanalysis: Applications to Speck, Simon and LEA ciphers. Scientia Sinica Informationis, 2025, 55(6): 1447–1470 (in Chinese with English abstract). [doi: 10.1360/SSI-2025-0024]

[17] Yang GQ, Zhu B, Suder V, Aagaard MD, Gong G. The simeck family of lightweight block ciphers. In: Proc. of the 17th Int'l Workshop on Cryptographic Hardware and Embedded Systems (CHES 2015). Saint-Malo: Springer, 2015. 307–329. [doi: 10.1007/978-3-662-48324-4_16]

[18] Gong RH. Key Recovery Attack on Lightweight Large-Block Ciphers using deep learning [MS. Thesis]. Xi'an: Xidian University, 2025

附中文参考文献

[14] 陈怡, 包珍珍, 申焱天, 于红波. 用于大状态分组密码的深度学习辅助密钥恢复框架. 中国科学: 信息科学, 2023, 53(7): 1348–1367. [doi: 10.1360/SSI-2022-0298]

[16] 申焱天, 陈怡, 于红波. 深度学习辅助密码分析的通用增强框架: 应用于 Speck, Simon 和 LEA 算法. 中国科学: 信息科学, 2025, 55(6): 1447–1470. [doi: 10.1360/SSI-2025-0024]

[18] 龚若涵. 轻量级大分组密码的智能化密钥恢复攻击 [硕士学位论文]. 西安: 西安电子科技大学, 2025.

作者简介

杨亚涛, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为密码学与信息安全, 后量子密码, 全同态加密, 密码协议和算法.

冉小玉, 硕士生, 主要研究领域为密码学与信息安全, 分组密码.

范修凤, 硕士生, 主要研究领域为密码学与信息安全, 分组密码.

王彩冰, 讲师, 主要研究领域为对称密码算法的设计与分析.