弱 AES 算法的设计与安全性分析

参赛队伍：黑盒三煞

指导老师：冯秀涛　曾祥勇

参赛选手：徐锐　陈新　马若为

参赛单位：中国科学院数学与系统科学研究所

时间：2025 年 XX 月 XX 日

目录

弱 AES 算法的设计与安全性分析

徐锐　陈新　马若为　冯秀涛 (指导老师)　曾祥勇 (指导老师)

中国科学院数学与系统科学研究所　2937519468@qq.com

摘要： 本文针对“设计并攻击一个弱化的 AES 算法”这一挑战，提出了一种系统性的解决方案。我们首先创新地设计了一个可参数化的弱 S 盒家族 $S_M(x) = M \cdot f(x)$，其中基础非线性函数 $f(x)$ 内嵌了极高的差分概率，而可逆线性层 M 则作为可优化的“攻击旋钮”。随后，我们采用一种两阶段自动化搜索策略：利用混合整数线性规划 (MILP) 在字节层面快速定位具有最少活跃 S 盒的 4 轮循环差分模式，再通过 SAT 求解器在比特层面进行精细化搜索，从而联合求解出一条高概率的循环差分路径以及完美匹配的最优 S 盒矩阵 M。基于该方案找到的 180 轮差分区分器，我们设计并实施了多组差分相结合的密钥恢复攻击。实验结果表明，该攻击方案在理论上能够成功攻破高达 182 轮的修改版 AES，其复杂度显著低于穷举搜索。本文完整地展示了从脆弱性设计、自动化路径搜索到密钥恢复攻击的全过程，并对攻击的理论边界进行了分析。

关键词：高级加密标准 (AES)；S 盒；差分密码分析；密钥恢复攻击；混合整数线性规划 (MILP)；SAT 求解器；循环差分特征

引言

本文档针对“通过修改 AES-128 算法的 S 盒和迭代轮数，设计一个安全性弱的分组算法”这一赛题展开研究。赛题的核心在于，在保持 AES 其他组件（如行移位、列混合）不变的前提下，设计一个非线性双射 S 盒，使其能够被高效攻击，并确定该攻击所能达到的最大轮数。

我们的核心解题思路是主动构造一个内嵌代数弱点的 S 盒，而非分析一个固定的密码。具体而言，我们设计了一个可参数化的 S 盒家族 $S_M(x) = M \cdot f(x)$，其中 $f(x)$ 是一个具有极高差分/线性概率的基础非线性函数，而可逆矩阵 M 则成为攻击者可以“定制”的自由参数。基于此设计，我们采用混合整数线性规划（MILP）和 SAT 求解器等自动化工具，协同搜索一条高概率的循环差分路径以及与之完美匹配的最优 S 盒。最终，我们利用找到的差分区分器，对修改后的 AES 算法实施了多组差分相结合的密钥恢复攻击。

通过该方案，我们成功地对一个 182 轮的修改版 AES 实现了攻击。本文档将详细阐述我们脆弱 S 盒的设计原理、差分路径的自动化搜索过程、完整的密钥恢复攻击方案以及最终的实验结果与分析。

一、 预备知识

1. AES 算法简介

高级加密标准（Advanced Encryption Standard，AES）是由美国国家标准与技术研究院（NIST）于 2001 年发布的对称分组密码算法。AES-128 是 AES 的一个具体实例，其分组长度和密钥长度均为 128 比特，采用标准 SPN 结构，被广泛应用于数据加密与信息安全领域。在后文中，无特殊说明情况下，我们直接将 AES-128 简记为 AES。

AES 的内部状态可以被描述为有限域 $\mathbb{F}_{2^8}$ 上的 4×4 矩阵。AES 的轮函数通过下面四个子部件构成:

- **字节替换**：对状态矩阵中的每个字节应用 S 盒进行非线性代换。
- **行移位**：循环移位状态矩阵的每一行，不同行移位数不同。

- **列混淆**：对状态矩阵的每一列进行线性变换，实现比特间扩散 (最后一轮省略)。
- **轮密钥加**：将当前状态与轮密钥进行按位异或。

密钥扩展 (Key Expansion) 是 AES 算法的一个重要步骤，其作用是将用户提供的 128 比特主密钥扩展为每一轮所需的轮密钥。

1. **初始分组**：将主密钥分为 4 个 32 比特的字，分别记为 W_0, W_1, W_2, W_3。
2. **迭代生成**：对 $i = 4, 5, \ldots, 43$，每个字 W_i 按如下方式生成：
   - 若 i 是 4 的倍数，$W_i = W_{i-4} \oplus \mathrm{T}(W_{i-1})$，其中 T 包括循环移位、S 盒替换及轮常量异或；
   - 否则，$W_i = W_{i-4} \oplus W_{i-1}$。
3. **T 操作具体流程**：
   (a) **循环移位**：将 32 比特字循环左移 8 位（即将第一个字节移到末尾）；
   (b) **字节替换**：对上述结果的每一字节应用 AES S 盒替换；
   (c) **轮常量异或**：将结果与轮常量 Rcon[$i/4$] 异或。
4. **最终输出**：上述过程共生成 44 个字，每 4 个字组成 1 组轮密钥，共 11 组，分别用于加密初始轮及 10 轮主过程。

2. 差分密钥恢复攻击

差分分析 (Differential Cryptanalysis) 是对称密码（尤其是分组密码）分析中最重要的攻击方法之一。该方法由 Biham 和 Shamir 于 1990 年代初提出，能够有效揭示加密算法在面对特定输入差分时输出差分的统计特性，从而帮助攻击者推断密钥。

差分分析的基本思想是：研究明文对之间的输入差分 (即两组明文的比特异或结果) 在经过若干轮加密后，对应的密文对之间的输出差分出现的概率。通过分析大量明文对的差分传播路径，攻击者可以找到高概率的差分特性，并利用这些特性进行密钥猜测和验证。

差分分析的常见流程包括：

1. 选取具有特定输入差分的明文对，获得对应的密文对；
2. 追踪和统计差分在加密过程中每一轮的传播情况，构造出高概率的差分路径（differential characteristic）；
3. 结合高概率路径和密文对，设计出针对部分轮密钥的猜测与验证，从而减少密钥搜索空间。

差分密钥恢复攻击的核心是寻找一个“差分特征”，即一个输入差分以较高的概率经过多轮加密后转变为某个特定的输出差分。攻击者通过分析大量的明密文对，筛选出满足该特征的密文对，从而推断出最后一轮的子密钥。

差分分布表 (DDT) 是一个用于分析 S 盒差分特性的重要工具。对于一个给定的 n-bit 的 S 盒，DDT 是一个 $2^n \times 2^n$ 的矩阵，其中第 α 行第 β 列的值表示输入差分为 α 时，输出差分为 β 的输入对的数量，即

$$DDT[\alpha][\beta] = \#\{x \in \mathbb{F}_2^n \mid S(x) \oplus S(x \oplus \alpha) = \beta\} \tag{1.1}$$

通过分析 DDT，攻击者可以确定哪些输入差分会导致高概率的特定输出差分，从而设计有效的差分攻击。

若某 S 盒的输入差分不为零，则称其为**活跃 S 盒**。在差分路径中，活跃 S 盒的数量直接影响路径的概率，因此寻找最小化活跃 S 盒的数量是构造高概率差分攻击路径的关键。

定义 1.1 (差分概率)**.** 设 $S : \mathbb{F}_2^n \to \mathbb{F}_2^n$ 为一个 n 比特 S 盒。对于给定的非零输入差分 $\alpha \in \mathbb{F}_2^n$ 和输出差分 $\beta \in \mathbb{F}_2^n$，其差分概率（*Differential Probability*）定义为：

$$DP_S(\alpha \to \beta) = \frac{|\{x \in \mathbb{F}_2^n \mid S(x) \oplus S(x \oplus \alpha) = \beta\}|}{2^n}$$

二、 弱 S 盒的设计与分析

我们的核心思想不是设计一个固定的、普适的弱 S 盒，而是构造一个可参数化的 S 盒家族。这个家族中的所有成员都内嵌了一个固有的代数弱点，但其具体的差分或线性

表现形式可以通过一个可逆矩阵 M 进行“调整”。在后续的攻击阶段，我们将通过搜索来确定最优的矩阵 M，使其 S 盒的弱点与 AES 的线性层（ShiftRows 和 MixColumns）完美配合，从而形成一条高效的攻击路径。

我们将 S 盒的构造分为两个部分：一个基础的、具有极简代数结构的非线性双射函数 $f(x)$，以及一个可逆的线性变换层 M。函数 $f(x)$ 是脆弱性的根本来源，而矩阵 M 则负责将这种脆弱性“旋转”和“对齐”，使其在多轮传播中最大化。我们不预先固定矩阵 M。相反，我们将 M 的选择视为攻击算法的一部分。攻击者的任务不仅是利用 S 盒的弱点，还包括“锻造”出最适合攻击的 S 盒。

我们提出的 S 盒 S_M 定义为一个复合函数，其形式为：

$$S_M(x) = M \cdot f(x) \tag{2.1}$$

其中，$x \in GF(2^8)$ 是输入字节，$f(x)$ 是 $GF(2^8)$ 上的基础非线性函数，M 是一个 8×8 的可逆二元矩阵。

1. 基础非线性函数 $f(x)$

为了引入结构性弱点，我们选择了一个结构最简单的非线性双射函数

$$f(x) = \begin{cases} 0, & x = 1; \\ 1, & x = 0; \\ x, & x \neq 0, 1. \end{cases} \tag{2.2}$$

计算 $f(x)$ 的 DDT，有

$$DDT[\alpha][\beta] = \begin{cases} 256, & \alpha = 0, 1, \\ 252, & \alpha = \beta \text{ 且} \alpha \neq 0, 1, \\ 4, & \alpha \oplus 1 = \beta, \\ 0, & \text{其他情况}. \end{cases} \tag{2.3}$$

该非线性函数的差分均匀度为 252/256, 非常接近理论上最大概率，所以我们认为从差分概率的角度, 该函数已经接近于最弱的 S 盒.

2. 可逆线性变换层 M

M 是一个 8×8 的二元可逆矩阵，它确保了最终的 S 盒 $S_M(x)$ 仍然是一个双射函数，且保持了基础非线性函数的差分特性。

为了攻击尽可能多的轮数, 我们希望可以找到一条尽可能长的高概率差分路径。然而，搜索长的高概率差分路径是一个困难的任务。观察到非线性双射函数的输入差分和输出差分相等的概率为 252/256，我们希望能够找到一条较短的循环差分路径，使得该路径的概率尽可能高。在 AES 算法线性操作固定的情况下，为使差分路径变化，只能通过对 S 盒的线性变换来实现。因此，我们要求对 M，存在一条高概率差分路径满足

$$\alpha_0 \to \alpha_1 \to \cdots \to \alpha_{r-1} \to \alpha_0. \tag{2.4}$$

这样的差分路径称可以被简单地拼接 k 次，从而构造出一条 $k\times r$ 轮的高概率差分区分器。在设计阶段，我们并不指定 M 的具体值。M 将作为攻击过程中的“自由参数”，与差分路径同时进行搜索求解。

我们之所以选择构造循环差分路径，其核心原因在于本次竞赛的目标是攻击尽可能多的轮数。对于一个上百轮的密码算法，搜索一条概率较高的、贯穿始终的单向差分路径在计算上是不可行的，其搜索空间过于庞大。相比之下，循环差分路径提供了一种高效的替代方案：我们只需找到一条轮数较短（例如 4 轮）但概率极高（即活跃 S 盒数量少）的循环特征，就可以通过简单地将其“拼接”或“迭代”多次，来构造一个覆盖任意长轮数的差分区分器。这种策略将一个极其复杂的长路径搜索问题，简化为了一个可控的、寻找最优短循环的优化问题。

三、 弱 AES 算法的差分攻击

本章详细阐述我们针对弱 AES 的密钥恢复攻击方案。该方案的核心是利用 SAT/SMT 求解器自动搜索高效的循环差分路径，并结合多组差分区分器来逐步恢复整个轮密钥。我们攻击的总轮数为 R_a 轮。

我们的攻击策略可以分解为两个主要阶段：

1. **差分路径搜索：**利用 SAT 工具搜索一条 R_d 轮的高概率差分路径，我们称之为“差分区分器”。这一步的目标是找到一个概率尽可能高（即复杂度中的 p 值尽可能小）的区分器。

2. **密钥恢复攻击：**基于找到的差分区分器，向前扩展一轮得到目标密文差分，向后拓展一轮得到目标明文差分，对 $R_a = R_d + 2$ 轮的修改版 AES 进行攻击。我们通过构造特定明文对，筛选密文对，并利用多组不同的区分器来迭代地约束并最终确定最后一轮的子密钥。

攻击复杂度与搜索目标 在展开具体搜索细节之前，我们首先对差分密钥恢复攻击的理论复杂度进行分析。攻击能否成功，关键在于能否找到至少一对遵循特定差分路径的“正确”明密文对，这直接决定了所需的数据量。

假设我们已经构造了一个 R_d 轮的差分区分器，其差分传播概率为 $P_{dist} = 2^{-p}$。在密钥恢复阶段，我们通常分别向前和向后各扩展一轮。然而，即便筛选出具有目标密文差分的密文对 (C_1, C_2)，也无法保证该对必然由区分器末端差分经最后一轮加密演化而来。事实上，一个特定密文差分可能由多种不同的输入差分以不同概率产生。

因此，有必要引入拓展轮的差分传播概率。设区分器末端差分 Δ_{out} 经最后一轮加密得到预设密文差分的概率为 $P_{final} = 2^{-p_1}$，而初始差分 Δ_{in} 经第一轮解密得到目标明文差分的概率为 $P_{begin} = 2^{-p_2}$。则一条完整的 $R_a = R_d + 2$ 轮差分路径（从目标明文差分到目标密文差分）成立的总概率为

$$P_{total} = P_{dist} \times P_{final} \times P_{begin} = 2^{-p} \times 2^{-p_1} \times 2^{-p_2} = 2^{-(p+p_1+p_2)}.$$

这意味着，为获得期望至少一个满足全路径的明密文对，需要加密并筛选约 2^{p+p_1} 组明文对。这即为攻击的数据复杂度。其中，p_1 与区分器末端差分 α_0 的活跃字节数 x 呈指数关系。尤其需要指出，由于我们选择的非线性函数差分分布表（DDT）在输入差分不为 0 和 1 时，输出差分仅有两种可能，且对应概率一个接近 1、一个接近 0。我们不能选用概率接近 1（即 $\frac{252}{256}$）的情况，因为此时对子密钥的过滤作用几乎消失。因此，最后一轮每个活跃 S 盒只能选用差分概率为 $\frac{4}{256} = 2^{-6}$，即 $2^{-p_1} = 2^{-6x}$。幸运的是，p_2

没有此限制，因为向明文方向扩展不涉及密钥猜测过滤，可选用概率为 $\frac{252}{256}$，那么不论初始差分的活跃字节个数是多少即 p_2 对总概率的影响远小于 p_1。

需要注意的是，活跃字节数越多，差分传播的总概率越低，导致 p_1 急剧减小。为了将总数据复杂度 2^{p+p_1} 控制在可行范围内，我们必须优先寻找不仅自身概率高（p 小），且起始与末端差分活跃字节数 x 极小（如 1 或 2）的循环差分路径。这正是我们自动化搜索算法的核心优化目标。

1. 循环差分路径与 S 盒的联合自动化搜索

传统差分分析中，攻击者通常是针对一个固定的密码算法来寻找最优差分路径。然而，在我们的竞赛场景下，S 盒本身是可变的，这引入了一个新的维度：我们不仅要寻找一条高效的路径，还要同时“锻造”出能够使这条路径成立的 S 盒。这个“路径-S 盒”的联合搜索空间极其庞大，手动分析几乎不可能完成。

为此，我们设计并实现了一个两阶段的自动化搜索策略。第一阶段使用混合整数线性规划（MILP）在字节层面快速筛选出具有优良循环结构的活跃 S 盒分布模式；第二阶段则在第一阶段结果的指导下，利用 SAT 求解器在比特层面进行精细化搜索，最终确定一条具体的、高概率的差分路径以及与之匹配的最优 S 盒线性层 M。

1.1. 基于 MILP 的最小活跃 S 盒个数搜索

在差分路径中，活跃 S 盒的个数决定了差分路径概率的上限，因此找到一条活跃 S 盒个数少的差分路径至关重要。因此，我们首先利用 MILP 在字节层面快速筛选出具有优良循环结构的活跃 S 盒分布模式。对于循环差分路径

$$\alpha_0 \to \alpha_1 \to \cdots \to \alpha_{r-1} \to \alpha_0. \tag{3.1}$$

其中 $\alpha_i \to \alpha_{i+1}$ 以概率 $p = 252/256$ 存在。我们使用混合整数线性规划（MILP）来解决这个问题，并将该模型交由高效的求解器 Gurobi 进行求解。首先，我们对 r 轮 AES 的字节级差分传播进行建模。定义 $x_{i,j} \in \mathbb{F}_2^n$ 是第 i 轮输入差分第 j 个字节的活跃状态，$y_{i,j} \in \mathbb{F}_2^n$ 是第 i 轮输出差分第 j 个字节的活跃状态，对差分传播过程进行建模：

1. **字节替换：**字节的活跃状态在 S 盒前后保持不变。

2. **行移位：**这是一个固定的字节置换，在模型中表现为变量索引的简单重映射。例如，第 i 轮行移位后的 (j,k) 字节的活跃状态，对应于行移位前的 $(j,(k+j) \pmod 4)$ 字节的活跃状态。

3. **列混淆：**这是建模的关键。由于列混淆的扩散分支数为 5，其差分传播遵循规则：对于任意一列，其输入活跃字节数与输出活跃字节数之和必须大于等于 5（如果输入不为 0）。这可以被精确地转化为一组线性不等式约束：

$$\sum_{j=0}^{3} x_{i,c+j}^{\text{in}} + \sum_{j=0}^{3} x_{i,c+j}^{\text{out}} \geq 5 \cdot \delta_i \tag{3.2}$$

   其中 x^{in} 和 x^{out} 分别代表 MixColumns 输入和输出的活跃状态变量，$\delta_i = 1$ 当且仅当该列有任意一个输入字节是活跃的。

4. **补充限制 1：**基于之前复杂度分析，我们补充一个限制条件，即 α_0 的活跃 S 盒个数不大于 2.

5. **补充限制 2：**我们强制要求第 r 轮的输出活跃模式与第 0 轮的输入活跃模式完全相同，即 $x_{r,j} = x_{0,j}$ 对所有 j 成立。

我们的目标是最小化整个循环特征中的总活跃 S 盒数量：

$$\min \sum_{i=0}^{r} \sum_{j=0}^{15} x_{i,j} \tag{3.3}$$

对轮数 r 从小到大进行搜索。我们将得到的一组具体的差分活跃模式称为 `ActivePattern`。

需要强调的是，对同一个 r，可能存在多组不同的 `ActivePattern`，甚至当 r 较大，比如 $r = 8$ 的时候，即使限制 α_0 的活跃字节不大于 2，也存在几千组不同的 `ActivePattern` 满足最小活跃 S 盒数量。我们要计算每一组不同的 `ActivePattern` 是否存在可行的差分特征，因此我们需要将这些不同的 `ActivePattern` 全部求出来。为了完成这一目标，我们使用了 Gurobi 求解器的 PoolSearchMode 函数，求解全部可行解。

1.2. 基于 SAT 的差分路径与矩阵 M 联合搜索

在获得 `ActivePattern` 后，我们利用 SAT 求解器同时寻找一个具体的差分路径 $\Omega = \{\alpha_0, \alpha_1, ..., \alpha_{r-1}\}$ 和一个能使该路径概率最高的线性矩阵 M。首先，对 r 轮循环差分路径

$$\alpha_0 \to \alpha_1 \to \ldots \to \alpha_{r-1} \to \alpha_0. \tag{3.4}$$

进行建模，定义第 i 轮差分的中间变量为 β_i，有

$$\alpha_{i+1} = M_{linear} \cdot \beta_i, \tag{3.5}$$

$$\beta_{i,j} = M \cdot \alpha_{i,j}, \tag{3.6}$$

$$\tag{3.7}$$

其中 M_{linear} 是 AES 中线性变换部分，$\alpha_{i,j}, \beta_{i,j}$ 分别表示第 i 轮差分及中间变量的第 j 个字节，定义 $m_{u,v}$ $(0 \le u, v \le 7)$，表示 8×8 矩阵 M 的元素。

要求差分路径满足 `ActivePattern` 对应的活跃状态，对于 `ActivePattern` 中标记为不活跃的字节，其对应的 8 个比特差分变量 Δx 被强制约束为 0。对于标记为活跃的字节，我们约束其 8 个比特差分中至少有一个为 1。

对公式 (3.5)，我们不直接在 SAT 求解器中描述矩阵乘法，而是采用**点集刻画**的思想。具体而言，上述矩阵方程被展开为 128 个独立的 XOR 方程，总结这 128 个方程对应的 anf(代数正规型) 种类，可以发现一共存在两种不同的 anf 函数，分别是。

1. $y = x_0 + x_1 + x_2 + x_3 + x_4$;

2. $y = x_0 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6$.

我们对这两个方程进行点集刻画，带入不同的变量，这样公式 (3.5) 的每一个这样的 XOR 方程都可以通过点集刻画高效地转换为一组 CNF 子句，从而精确地约束了差分在 **L** 变换下的传播行为。

公式 (3.6) 是对 S 盒 $S_M(x)$ 差分变换的描述。根据我们 S 盒的设计 $S_M(x) = M \cdot f(x)$，以及我们只利用其高概率差分传播 $\alpha \to \alpha$ 的策略，对于一个输入差分为 α 的函数 f，我们强制要求其输出差分 β 必须满足 $\beta = M \cdot \alpha$。进而这个关系可以展开

为 8 个涉及变量乘积的异或方程:

$$\beta_i = \oplus_{j=i}^{7}(m_{i,j} \wedge \alpha_j) \tag{3.8}$$

这 8 个方程对应相同的 anf 函数:

$$y = x_0 * x_1 + x_2 * x_3 + \ldots + x_{14} * x_{15}. \tag{3.9}$$

对于这个函数，我们采用了两种不同的方法进行处理，一种是直接点集刻画这个函数，它对应的是 17 维点集刻画。第二种方法是引入新的中间辅助变量 z_0, z_1，将式 (3.9) 分解为:

$$z_0 = x_0 * x_1 + x_2 * x_3 + x_4 * x_5 + x_6 * x_7; \tag{3.10}$$

$$z_1 = x_8 * x_9 + x_{10} * x_{11} + x_{12} * x_{13} + x_{14} * x_{15}; \tag{3.11}$$

$$\beta_0 = z_0 + z_1. \tag{3.12}$$

它们分别对应 2 类 anf，第一类包含 9 个变元，第二类包含 3 个变元。

我们分别对两个方法进行点集刻画然后测试效率，最后结果显示采用第二种方法 Cadical 求解器可以更快得到结果，因此，我们统一使用第二种方法进行建模。

对于可逆矩阵 M，由于可逆矩阵这个性质难以直接刻画，而且 $\mathbb{F}_2$ 上 8×8 的可逆矩阵数量庞大。因为，我们在这一步对 M 加强限制条件，令 M 是一个单位上三角矩阵，从而简化 SAT 模型。

将上述所有约束输入到 SAT 求解器中，求解得到了我们所需的循环差分路径 Ω 和对应的 S 盒线性层矩阵 M。我们小组所采用的 SAT 求解器为 Cadical。

2. 不同轮数结果说明

当 $r = 1$ 时，Cadical 显示有解的最小活跃 S 盒个数为 8。因此，我们可以直接得到一个任意轮循环差分迹活跃 S 盒个数上界。但是这个路径对应的 α_0 活跃 S 盒个数远大于 2，不符合我们的要求。当 $r = 2, 3$ 的时候，我们没有找到小于这个上界的解。当 $r = 4$ 的时候，Cadical 显示有解的最小活跃 S 盒个数为 8，对应的 α_0 活跃 S 盒个

数正好为 2。然后我们固定得到 8×8 矩阵 M，求解是否存在其他差分可以满足条件。最后我们得到 8 条不同的循环差分迹，这 8 条路径正好包含了 α_0 的全部 16 个字节。因此，我们分别使用这 8 个循环差分迹进行差分密钥恢复攻击，我们可以得到完整一轮的全部子密钥。我们将在下一章给出具体差分特征以及使用的仿射矩阵。

该路径的每一个活跃 S 盒都以 $p_s = 252/256$ 的概率传播，因此我们 4 轮循环特征的总概率为 $p = (252/256)^{30} \approx 2^{-0.68}$。将此特征循环 k 次，得到了概率为 $2^{-0.68\times r}$ 的 $R_d = 4k$ 轮差分区分器。

此外，为了确保我们所选方案的优越性，我们基于同样的模型，对起始活跃字节数不大于 2 的 5 轮和起始活跃字节数为 1 的 6 轮、7 轮和 8 轮循环差分路径也进行了搜索。实验结果表明，这些更长轮数的循环路径所能达到的最小活跃 S 盒总数，均劣于我们找到的 4 轮 30 个活跃 S 盒的结果。因此，我们有理由相信，该 4 轮循环特征是在 8 轮以内最优的循环结构，为后续构建高效的长轮数区分器奠定了坚实的基础。

3. 弱 AES 算法的密钥恢复攻击

我们的目标是攻击一个总轮数为 $R_a = 4k + 2$ 的弱 AES 算法。攻击的核心是利用一个精心构造的 $R_d = 4k$ 轮差分区分器，向前向后分别拓展一轮得到目标明密文差分，并在最后一轮进行密钥恢复。该 $4r$ 轮区分器通过拼接 k 次 4 轮循环特征（共 $30r$ 个活跃 S 盒）得到，其总概率为 $P_{dist} = (252/256)^{30k}$。

为了极大地提高信噪比，我们筛选那些能以特定低概率 $p_1 = (4/256)^2$ 生成的目标密文差分 ΔC。另一方面，由于 α_0 的两个活跃字节处于不同行，因此，目标明文差分的活跃状态是固定的，选择其中概率最大的传播路径，则生成目标明文差分 Δ_P 的概率为 $p_2 = (252/256)^8$。因此，一个满足我们完整差分路径的“正确对”出现的总概率为：

$$P_{total} = P_{dist} \times p_1 \times p_2 = (252/256)^{30k+8} \times 2^{-12}. \tag{3.13}$$

攻击流程如下：

1. **数据收集与筛选：** 构造足够数量（略多于 P_{total}^{-1}）的、具有输入差分 α_0 的明文对。加密后，筛选出满足目标密文差分 ΔC 的密文对 (C_1, C_2)。由于我们收集了足量的数据，预期会得到一小撮候选密文对。

2. **密钥恢复：**对每一个通过筛选的候选密文对，我们猜测最后一轮子密钥中相关的字节（例如，通过多组攻击，每次猜测 2 个字节，即 2^{16} 种可能）。用猜测的密钥进行一轮逆向解密，并检查中间差分是否与区分器的末端差分 α_0 一致。由于候选密文对的数量极少，这一步的计算成本远低于数据收集阶段。

3. **密钥确定：**我们精心设计了 8 组不同的差分区分器，每组针对不同的密钥字节。通过综合 8 次攻击的结果，可以唯一确定最后一轮的全部 128 位子密钥。

4. 攻击复杂度分析

本节的复杂度分析建立在一个关键假设之上：我们所找到的 4 轮循环差分特征在 AES-128 标准密钥编排下是真实有效的，即它不是一个“奇异特征”[4]。基于此假设，我们攻击方案的总复杂度主要由数据收集和筛选的工作量决定。为了恢复完整的 128 位密钥，我们设计了 8 组独立的差分攻击，每组攻击针对不同的密钥字节。

我们攻击方案的总复杂度主要由数据收集和筛选的工作量决定。为了恢复完整的 128 位密钥，我们设计了 8 组独立的差分攻击，每组攻击针对不同的密钥字节。

攻击的总复杂度（Complexity）可以由以下通用公式表示：

$$\text{Complexity} \approx N_{attacks} \times c \times P_{total}^{-1} \tag{3.14}$$

其中：

- $N_{attacks}$: 恢复全密钥所需的独立攻击组数，即进行了多少种不同的区分攻击。在我们的方案中，$N_{attacks} = 8$。
- c: 一个小的常数因子，代表为保证高成功率而进行的“过采样”，以及对筛选出的少量候选密文对进行密钥验证的计算成本。我们设定 $c = 16 = 2^4$。
- P_{total}: 单次攻击中，一个“正确”明密文对出现的总概率。对于一个 $R_a = 4k + 2$ 轮的攻击，该概率为：

$$P_{total}(k) = (\frac{252}{256})^{30k+8} \times 2^{-12}. \tag{3.15}$$

将这些参数代入，我们的攻击复杂度模型简化为：

$$\text{Complexity}(k) \approx 8 \times 16 \times P_{total}(k)^{-1} = 2^7 \times P_{total}(k)^{-1}. \tag{3.16}$$

在下一章中，我们将使用此模型对具体攻击轮数进行性能评估。

5. 模型局限性与“奇异特征”风险

值得注意的是，我们的整个攻击流程与标准差分分析一致，均基于以下两个核心假设:

1. **随机等价性假设（stochastic equivalence hypothesis）**：假定差分概率对于不同密钥的变化极小；
2. **马尔可夫密码（Markov cipher）假设**：假定被攻击的密码原语为马尔可夫密码，其轮密钥彼此独立且均匀分布，与具体的密钥扩展结构无关。

然而，这两项假设本身也蕴含潜在的理论风险。正如 Liu 等人在其关于“奇异特征”（Singular Characteristics）的开创性工作 [4] 中所指出的，某些理论上概率极高的差分路径，可能由于与实际密钥编排算法生成的轮密钥序列存在不可避免的代数冲突，而对所有主密钥均不成立。此类路径被称为“奇异特征”，基于这些特征的攻击将注定失败。

因此，我们必须坦诚地指出：尽管我们找到的 180 轮差分区分器在数据路径层面具有极高的理论概率，但其仍然存在成为“奇异特征”的风险。为验证该差分特征在 AES 密钥编排下的真实有效性，我们可以从理论和实验两个角度进行论证。理论方面，可采用 [4] 中的算法 1，寻找能够使该差分特征成立的主密钥。实验方面，则可多次重复攻击过程（例如 100 次），统计获得正确密钥的次数，进而估算攻击的实际成功概率。

受限于时间，我们目前仅对简化轮数的密码算法进行了重复性实验分析。针对完整轮数的验证以及理论上的奇异性分析，将作为我们后续研究的重点方向。在随后的复杂度分析中，我们的结论均建立在“该差分路径不是奇异特征”这一假设基础之上。

四、 实验结果与分析

为了验证我们所提攻击方案的有效性，并探寻其所能达到的最大攻击轮数，我们开展了系统性的自动化搜索与分析。本章将详细介绍实验配置、展示搜索得到的关键成果（包括差分循环迹与最优 S 盒线性层），并对攻击 182 轮修改版 AES 的性能进行分析。

1. 实验环境与自动化搜索配置

离线的自动化搜索是本次研究中计算量最大的部分，所有搜索和分析实验均在以下环境中进行：

- **硬件平台：**高性能计算服务器 (Intel Xeon(R) Gold 6258R CPU@ 2.70GHz ×112)。
- **软件环境：**操作系统为 Ubuntu 20.04，攻击代码和模拟器采用 C++17 编写，并由 g++ 11.2 编译。
- **关键工具：**我们使用 `Gurobi 10.0` 作为 MILP 求解器，用于搜索字节级的循环特征；使用 `CaDiCaL` 作为后端的 SAT 求解器，用于进行比特级的差分路径与 S 盒矩阵的联合搜索。

2. 自动化搜索核心结果

我们的两阶段自动化搜索策略取得了预期的成功，不仅发现了高效的循环差分特征，还同时确定了与之完美匹配的最优 S 盒。

2.1. 4 轮循环差分特征

在第一阶段，MILP 模型在对 AES 的字节级差分传播进行建模后，成功找到了一个极优的 4 轮循环差分特征。该特征在 4 轮内总共仅包含 **30 个活跃 S 盒**，这是保证长轮数攻击概率不会过快衰减的基础。该特征的活跃 S 盒分布模式（我们称之为 `ActivePattern`）如下表1所示，其中'1' 代表活跃字节，'0' 代表非活跃字节。该模式满足循环条件，即第 4 轮 MixColumns 变换后的活跃模式与第 0 轮的初始模式完全相同。

表 1: MILP 找到的 4 轮 30 活跃 S 盒循环特征模式

Round 0	**Round 1**	**Round 2**	**Round 3**
0 0 0 0	1 0 1 0	0 1 0 1	1 0 1 0
0 0 0 0	1 0 1 0	1 1 1 1	0 1 0 1
1 0 1 0	1 0 1 0	1 0 1 0	1 0 1 0
0 0 0 0	1 0 1 0	1 1 1 1	0 1 0 1

2.2. 最优线性层与 4 轮循环差分路径

在第二阶段，以 `ActivePattern` 为指引，我们的 SAT 模型在比特层面进行了精细化搜索。求解器不仅找到了一条遵循该模式的具体差分路径，还同时给出了使该路径成立的最优 S 盒线性层矩阵 M。我们找到的一个满足所有约束的可逆上三角矩阵 M 如下所示：

$$M = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.1}$$

由于篇幅限制，我们在正文中仅展示 8 条循环差分迹中一条的具体表示（16 进制下），全部 8 条差分迹则在补充文件给出:

```
alpha_0: 0x00, 0x00, 0x4f, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x4f, 0x00, 0x00, 0x00, 0x00, 0x00;
alpha_1: 0x4f, 0xd1, 0x9e, 0x4f, 0x00, 0x00, 0x00, 0x00, 0x4f, 0xd1, 0x9e, 0x4f, 0x00, 0x00, 0x00, 0x00;
alpha_2: 0x00, 0xf6, 0x68, 0x4f, 0x27, 0xf6, 0x00, 0x4f, 0x00, 0xf6, 0x68, 0x4f, 0x27, 0xf6, 0x00, 0x4f;
alpha_3: 0x26, 0x00, 0xf7, 0x00, 0x00, 0x9f, 0x00, 0x01, 0x26, 0x00, 0xf7, 0x00, 0x00, 0x9f, 0x00, 0x01.
```

该 4 轮循环特征包含 30 个活跃 S 盒，且每个 S 盒的差分传播都遵循高概率的 $\alpha \to M\alpha$ 模式，其概率为 $p_s = 252/256$。因此，该 4 轮基础特征的概率为 $(252/256)^{30} \approx 2^{-0.68}$。

3. 攻击性能分析与轮数上限

在本节中，我们将应用第四章建立的复杂度模型（公式 3.16）来评估我们攻击方案的实际性能，并推断其理论极限。

- **实验验证边界 (182 轮):** 我们的主要攻击目标为 $R = 182$ 轮，对应模型中的 $r = 45$。首先，我们计算单次攻击的成功概率 P_{total}:

  $$P_{total}(45) \approx (252/256)^{30\times 45+8} \times (4/256)^2 = (252/256)^{1358} \times 2^{-12} \approx 2^{-30.85} \times 2^{-12} = 2^{-42.85}$$

  根据公式 3.16，总攻击复杂度为:

  $$\text{Complexity}(45) \approx 2^7 \times (2^{-42.85})^{-1} = 2^{49.85}$$

  该复杂度远低于 2^{128} 的穷举搜索门槛，证明了我们针对 182 轮算法的攻击方案在理论上是可行且高效的。

- **当前技术与理论边界:** 利用复杂度模型，我们可以外推出在不同算力限制下的最大攻击轮数。

下表 2 总结了基于我们攻击模型的三个关键性能边界。

表 2: 不同技术条件下的攻击边界分析

攻击边界	**最大轮数**	**攻击复杂度 (约)**
实验验证边界	182 轮	$2^{49.85}$
当前技术边界	262 轮	$\approx 2^{64}$
理论攻击边界	638 轮	$\approx 2^{128}$

实验验证边界具体表示本次研究中实际设计并验证的攻击方案；当前技术边界表示在当前主流高性能计算能力下的可行攻击上限；理论攻击边界表示攻击复杂度达到穷举搜索门槛，为该方法的理论极限。

分析结果表明，我们实验验证的 182 轮攻击的复杂度（约 $2^{49.85}$）处于一个可接受的理论范畴内。更重要的是，它证明了我们的攻击方法是成立的，并且显著优于穷举

搜索。因此，我们得出最终结论：**我们所设计的攻击方案，在理论上可成功应用于高达 638 轮的修改版 AES，而本次研究所能达到的最大攻击轮数为 182 轮**。这一结果充分展示了我们“S 盒与差分路径协同搜索”策略的强大威力。

4. 攻击成功概率

根据 [4]，我们知道即便我们找到一条具体的高概率差分路径，在现实分析中，任然存在攻击失败的可能，特别是当轮数较大的时候。这是因为差分分析都是基于密码函数马尔科夫链假设以及忽略轮密钥加对内部状态的影响导致的。因此，我们有必要检验我们的差分分析成功概率，即在 100 次重复实验中，大概有多少次可以成功找到正确子密钥。

由于 182 轮每次攻击耗时太大，因此我们不得不对约减轮数算法进行重复实验。我们对 102 轮密码算法进行重复实验，通过实现显示，其中有 99 次实验可以找到唯一正确的子密钥。

五、 结论

本文围绕“通过修改 AES 的 S 盒以攻击最大轮数”这一核心挑战，展开了系统性的研究与实践。我们不仅成功地设计出一种新颖的、可被攻击者利用的 S 盒，还基于此实现了一套完整且高效的密钥恢复攻击方案，并最终确定了该攻击在理论、当前技术以及我们实验环境下的三个不同层次的攻击边界。

1. 主要工作与贡献回顾

本文的主要贡献可以总结为以下几点：

1. **提出了一种可参数化的脆弱 S 盒设计方法：**我们创新性地将 S 盒设计为基础非线性函数与可逆线性矩阵的复合 $S_M(x) = M \cdot f(x)$。这种“分离与调优”的设计哲学，使得攻击者可以将 S 盒的弱点“对齐” AES 的线性层，从而主动构造出高效的攻击路径。

2. **实现了基于 SAT/MILP 的自动化攻击路径搜索方案：**我们利用两阶段自动化工具，高效地搜索到了一个理论上概率极高的 4 轮 30 活跃 S 盒循环差分特征，并基于此构建了多轮密钥恢复攻击。

3. **在理论上验证了攻击的有效性：**通过复杂度分析，我们证明了在“路径有效”的假设下，对一个 **182 轮**的修改版 AES 实施密钥恢复攻击是可行的，其复杂度远低于穷举。

2. 局限性与未来工作

本研究最主要的局限性在于，我们的自动化差分路径搜索模型未将 AES 密钥编排算法的约束包含在内。正如相关研究所指出的 [4]，这使得我们找到的高概率路径存在是“奇异特征”的风险，即可能对所有密钥都无效。因此，我们当前攻击方案的成立，是建立在“所用路径非奇异”这一重要假设之上的。

未来的关键一步，是将密钥编排完整地建模并整合进我们的 SAT 求解器中，进行一次更全面的、包含数据路径、S 盒结构和密钥生成的三方联合搜索。这将从根本上消除“奇异特征”的风险，确保攻击路径的真实性。我们相信，这样的改进将使我们的攻击方案更加严谨和强大，并可能进一步优化攻击的轮数和复杂度。

3. 核心结论与意义

本研究的核心结论是：AES 的安全性对其 S 盒的优良密码学属性（如高非线性度、低差分均匀性、高代数次数）具有高度的敏感性。即使 S 盒仅仅在一个维度上被显著削弱，整个密码系统的安全性也会在现代密码分析技术面前急剧下降。

我们的工作从一个逆向的视角——“如何有效地破坏安全”，直观且深刻地揭示了 AES 设计者在选择 S 盒时的深思熟虑。它不仅是一次成功的密码攻击实践，更是一次对 AES 安全基石的“压力测试”，反过来印证了标准 AES 设计的卓越与稳健。

综上所述，本文完成了一次从脆弱性设计到密码分析实践的完整探索，为理解分组密码的核心安全机制提供了有价值的参考和实例。

参考文献

[1] National Institute of Standards and Technology (NIST), *FIPS PUB 197: Advanced Encryption Standard (AES)*, 2001.

[2] E. Biham and A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.

[3] M. Matsui, *Linear Cryptanalysis Method for DES Cipher*, In Advances in Cryptology —EUROCRYPT '93, pp. 386–397, 1994.

[4] Y. Liu, W. Zhang, B. Sun, et al. *The phantom of differential characteristics*. Des. Codes Cryptogr. 88, 2289–2311 (2020).