目 录

[目 录 1](#_Toc193097630)

[1. 概述 1](#_Toc193097631)

[**1.1 项目背景** 1](#_Toc193097632)

[**1.2 研究目标** 1](#_Toc193097633)

[**1.3 研究内容** 2](#_Toc193097634)

[2. 国内外研究现状 4](#_Toc193097635)

[3.一种面向比特的杂凑密码设计 5](#_Toc193097636)

[**3.1 自动化面向比特的杂凑密码设计框架** 6](#_Toc193097637)

[**3.2 轮函数线性层建模** 8](#_Toc193097638)

[**3.3 轮函数非线性层建模** 10](#_Toc193097639)

[**3.4 密码实现建模** 11](#_Toc193097640)

[**3.5 安全性分析建模** 12](#_Toc193097641)

[**3.6 性能分析建模** 14](#_Toc193097642)

[4. 面向比特的杂凑密码设计准则 15](#_Toc193097643)

[**4.1 结构设计** 15](#_Toc193097644)

[**4.2 线性层设计** 18](#_Toc193097645)

[**4.3 非线性层设计** 18](#_Toc193097646)

[**4.4 抗攻击能力准则** 19](#_Toc193097647)

[**5 面向比特的杂凑密码设计关键技术** 20](#_Toc193097648)

[**5.1 自动化构建差分分布表** 21](#_Toc193097649)

[**5.2 对差分均匀度及线性度的约束** 22](#_Toc193097650)

[**5.3 对差分均匀度及线性度的频率约束** 23](#_Toc193097651)

[**5.4 对BIBO模式个数的约束** 24](#_Toc193097652)

[**5.5 对S盒本身的约束** 26](#_Toc193097653)

[5. 安全性分析 26](#_Toc193097654)

[6. 成果展示 27](#_Toc193097655)

[7. 总结展望 27](#_Toc193097656)

[8. 参考文献 27](#_Toc193097657)

一种面向比特的杂凑密码设计研究课题

2025年度研究报告

1. 概述

**1.1 项目背景**

在现代信息社会中，数据的完整性和不可篡改性是确保信息安全的重要基础。作为实现数据完整性保护的重要工具，杂凑密码在众多信息系统和安全协议中得到了广泛应用。随着信息技术的快速发展，数据量的剧增和新型攻击手段的不断涌现，对杂凑密码的吞吐率和安全强度提出了更高的要求。然而，传统杂凑密码的设计和分析方法通常依赖于人工经验和试错过程，设计效率较低，难以全面保障其安全性和性能的平衡。因此，如何提高杂凑密码的自动化设计能力，增强其抗攻击能力，成为当前密码学领域亟待解决的关键问题。

近年来，自动化建模技术的发展为杂凑密码的设计和分析带来了新的机遇。通过引入符号执行、约束满足技术（如SAT/SMT求解器）和数学规划等自动化工具，密码设计者可以更加系统、全面地探索设计空间，加速杂凑密码的开发进程。同时，这些技术还可用于自动化分析，帮助密码分析人员有效识别潜在的弱点，提高分析效率和精确性。

在该背景下，本项目旨在利用当前杂凑密码设计和分析技术的最新进展，结合自动化建模方法，提升杂凑密码的吞吐率和安全强度。项目将重点研发杂凑密码的自动化设计算法，以实现高效、强安全性的杂凑密码构建。同时，项目还将发展几种典型的杂凑密码分析技术，包括对SHA-3类杂凑密码的中间相遇原像分析技术，以及基于字的SPN结构杂凑密码的Rebound自动化分析技术。此外，项目将探索智能化的杂凑密码分析方法，推动杂凑密码的自动化设计与分析技术的进一步发展，为下一代杂凑密码的设计和实现奠定基础。

**1.2 研究目标**

本项目的研究目标为：针对军事杂凑密码算法的自动化分析与设计问题，充分利用目前杂凑密码分析技术以及自动化建模技术，以提升杂凑密码的吞吐率和安全强度为目标，研发杂凑密码的自动化设计算法，发展杂凑密码的几种典型分析技术，推动杂凑密码的自动化设计和分析技术的发展。

本年度的研究目标包括：

（1）研究基于字的SPN结构的杂凑密码的Rebound（半）自由起始碰撞分析的自动化算法，给出轮数达到现有分析结果、时间或空间复杂度低于现有结果Rebound分析结果，建立自动化分析的C或Python程序代码资源库，程序代码遵循统一制定的接口规范。

（2）研究SHA-3类杂凑密码的中间相遇原像分析方法的自动化建模技术，给出轮数达到现有分析结果、时间或空间复杂度低于现有结果的原像分析结果，建立自动化分析的C或Python程序代码资源库，程序代码遵循统一制定的接口规范。

本年度的研究目标完成情况如下：

1）研究基于字的SPN结构的杂凑密码的Rebound分析方法的自动化建模技术，实现了基于字的SPN结构的杂凑密码的Rebound（半）自由起始碰撞分析自动化平台。

2）研究SHA-3类杂凑密码的中间相遇原像分析的自动化建模技术，实现了SHA-3类杂凑密码的中间相遇原像分析自动化平台。

**1.3 研究内容**

杂凑密码的设计研究内容如下。

（1）单向函数和杂凑密码组件设计准则研究

针对单向函数的抗碰撞、不可逆等密码特性，研究提出单向函数的设计准则和指标要求，研究置换的自动化设计方法，对于非线性组件，研究其安全属性的自动化建模方法，选出具有高非线性度、低差分均匀性和高代数次数，且差分分布表（DDT）和线性近似表（LAT）性质良好的非线性组件；对于线性组件，研究其安全属性的自动化建模方法，使得线性组件的分支数达到最大、实现性能良好。

（2）杂凑密码自动化设计技术研究

为了设计出高性能、高安全性的杂凑密码，研究杂凑密码的自动化设计技术，研究基于MILP或SAT自动化建模选取杂凑密码的非线性组件和线性组件参数的方法，结合自动化搜索和检验等技术，研究杂凑密码整体安全指标和组件安全指标的检验方法。设计主要遵循的原则是简单化，只有在必要的时候，比如为了抵抗某种分析，才增加相应的组件。

2. 国内外研究现状

杂凑函数在现代密码学中占据核心地位，其设计与发展一直是国内外密码学研究的重点。国际上，Merkle 和 Damgård 在 20 世纪 80 年代独立提出了 Merkle-Damgård（MD）结构，成为最早的迭代杂凑函数结构之一。这一结构能够将抗碰撞的压缩函数扩展为抗碰撞的杂凑函数，从而简化了杂凑函数的设计问题，使得后续的杂凑算法多基于这一框架构造。Rivest 在 1990 年的 CRYPTO 会议上提出了 MD4 算法，随后又发展出 MD5、HAVAL、SHA-1、SHA-2、RIPEMD 系列等基于 MD 结构的杂凑函数。这些算法被广泛应用于密码学、数据完整性验证、数字签名等领域。然而，随着密码分析技术的发展，MD 结构暴露出一些安全性弱点，例如长度扩展攻击、二次碰撞攻击、多碰撞攻击和长消息第二原像攻击，这促使研究人员对其进行改进和优化。

针对 MD 结构的安全问题，学者们提出了多种改进方案。例如，Shoup 结构和Zipper 结构通过引入更复杂的运算方式来提高安全性，但这些方法往往增加计算复杂度，降低了实际应用中的效率。2005 年，Lucks 在ASIACRYPT 会议上提出了双管道结构（Double Pipe）和宽管道结构（Wide Pipe），用于抵抗生日攻击和多碰撞攻击。RIPEMD 系列算法采用典型的双管道结构，并联两个相同尺寸的窄管道压缩函数，提高了安全性和抗碰撞能力。然而，2015 年 Laurent 和 Wang 在 EUROCRYPT 会议上指出，这种方法可能降低杂凑函数抵抗原像攻击的能力。此外，Biham 和 Dunkelman 在 2006 年提出了 HAIFA 结构，通过在压缩函数输入中引入随机盐值（Salt）和已压缩比特数 来增强安全性。BLAKE 算法作为 SHA-3 最终轮的候选算法之一，就是基于 HAIFA 结构设计的。

近年来，国际上越来越多的研究关注海绵结构（Sponge Construction）。2007 年，Bertoni 等人在 EUROCRYPT 会议上提出了这一基于大状态置换的杂凑函数结构，Keccak 团队进一步扩展并提出了 Duplex 结构，最终 Keccak 成为 SHA-3 标准算法。此外，许多轻量级杂凑函数，如 QUARK、Photon 和 SPONGENT，均基于海绵结构构造，旨在满足物联网、智能卡等受限环境的安全需求。

除了经典结构的演变，基于分组密码算法的杂凑函数也受到关注。例如，ISO/IEC 10118-3 标准算法 Whirlpool，以及 SHA-3 竞赛的候选算法 ECHO、Grøstl 和 Skein，均是采用分组密码设计的杂凑函数。这些算法利用现有的强密码块加密技术，提供了一定的安全性保障，同时也增加了计算复杂度。

在国家密码标准方面，各国纷纷推出自主设计的杂凑函数算法。2010 年，我国国家商用密码管理办公室推出了商用杂凑函数标准 SM3，并于 2012 年确立为国家密码行业标准 GM/T 0004-2012。SM3 采用类似 SHA-2 的结构，但针对特定攻击做了优化，提高了安全性和抗分析能力。俄罗斯在 2012 年确立了新的国家杂凑密码标准 GOST R 34.11-2012，而韩国则在 2000 年推出了 HAS-160 算法，专门用于本国的数字签名标准。这些国家级杂凑函数标准的制定，表明各国在密码安全领域的自主研发能力不断增强，以适应本国的安全需求。

随着物联网（IoT）、云计算和人工智能技术的发展，轻量级杂凑函数 逐渐成为研究热点，以满足资源受限设备的安全需求。Shamir 在 2008 年 FSE 会议上提出了 SQUASH 轻量级 MAC 算法，随后 Bogdanov 等人在 CHES 会议上推出 H-PRESENT-128、Quark 和 SPONGENT 等轻量级杂凑算法。我国学者吴文玲等人也在 2013 年提出了基于海绵体结构的轻量级杂凑函数 LHash，致力于提升安全性与计算效率的平衡。近年来，基于 细胞自动机（Cellular Automata） 的轻量级杂凑函数如 L-CAHASH 也在 UNet 2017 会议上被提出，为新型杂凑函数结构提供了新的研究方向。

总体来看，国际上杂凑函数的研究方向正在从传统 MD 结构向更安全、更高效的结构转变，海绵结构、HAIFA 结构、双管道结构等新兴架构逐步取代了 MD 结构，并在 SHA-3 及后续算法中得到了应用。同时，各国也在积极制定自主杂凑函数标准，以应对国家安全需求。随着新兴应用场景的增加，轻量级杂凑函数设计也成为密码学研究的重要领域，推动着未来杂凑函数算法的进一步发展。

3. 面向字节的杂凑密码设计准则

**3.1 结构设计**

JH 结构由Hongjun Wu 等人提出，作为一种创新的算法设计架构，在密码学领域展现出卓越性能，被应用于构建高效且安全的杂凑及分组密码算法。​

JH 结构具备诸多显著优点。首先，在存储与计算资源利用上，其新型压缩函数架构通过恒定密钥的分组密码构建压缩函数，避免了复杂的密钥调度及中间变量引入，使得计算过程简洁直接。其次，基于广义 AES 设计方法论，JH 结构能够从小型组件便捷构建大规模分组密码，展现出极高的设计灵活性。这种灵活性体现在可根据不同应用场景，调整维度与组件配置，满足多样化的安全与性能需求。最后，JH 结构在安全性方面表现突出，其压缩函数通过大量活跃 S 盒设计，有效抵御差分攻击，并且安全性分析相对简明，便于根据实际应用需求在效率与安全性间实现精准权衡。​

JH 结构在处理消息时，融合了独特的压缩函数架构与广义 AES 设计流程。在输入层面，消息被按照特定规则处理，进而参与后续关键运算阶段。其工作过程主要分为以下几个关键阶段：​

(1) 输入处理阶段​

在 JH 结构处理消息的起始阶段，对于输入的消息，首先会根据广义 AES 设计方法论进行数据组织。输入比特被划分为​个元素，并构成 d 维数组。例如在八维广义 AES 架构中，将 1024 比特输入划分为 256 个 4 比特元素，构成八维数组。这一过程使得输入数据能够以一种结构化、便于后续处理的形式存在，为后续的轮函数操作奠定基础，确保数据在不同维度上能够被有序且高效地处理，提升整体算法对不同规模数据的适应性与处理能力。​

(2) 压缩函数运算阶段​

JH 结构采用新型压缩函数架构，此阶段以恒定密钥的分组密码为基础构建压缩函数。以杂凑计算场景为例，2m 比特的杂凑值 H(i−1)与 m 比特的消息块 M(i)被压缩为 2m 比特的 H(i)，消息摘要长度最大为 m 比特。由于分组密码的密钥设定为常量（即置换操作），中间过程未引入额外变量，使得针对差分攻击的安全性分析更为便捷，同时不截断分组密码输出的设计显著提升了计算效率。在每一轮运算中，轮函数通过类 AES 行循环移位技术保持结构一致性，并且在特定维度方向应用 MDS 码，将输入数据充分混合，不断更新内部状态，实现对消息的深度压缩与特征提取。​

(3) 安全性强化阶段​

JH 结构的安全性保障贯穿整个运算过程。在压缩函数运算阶段，每个消息块长度为 64 字节，需通过含 10752 个 16 位 S 盒的 42 轮压缩函数处理。经分析，压缩函数中任意差分路径均涉及超过 700 个活跃 S 盒，大量活跃 S 盒的存在有效确保了算法对差分攻击的强抵抗能力。同时，通过广义 AES 设计方法论将 AES 设计范式推广至高维空间，基于小型组件构建大规模分组密码的方式，进一步从结构设计层面增强了整体安全性，使得算法在面对各类潜在攻击时具备坚实的防护能力。​

(4) 输出生成阶段​

经过前面多个阶段的运算处理，最终生成符合要求的输出结果。依据前期设定的规则与运算结果产生输出值。输出结果在长度、格式等方面均满足对应应用场景的标准与需求，为后续的信息存储、传输或其他相关操作提供可靠的数据形式。

**3.2 线性层设计**

线性层的主要目标是提供快速扩散能力，使得输入的微小变化能够迅速传播到整个状态，以增强算法的抗差分攻击和抗线性攻击能力。为了实现这一目标，线性层的设计应遵循以下原则：

高扩散性：线性层应能够在较少轮数内实现全扩散，即所有字节均受到输入变化的影响，从而避免攻击者通过差分特性进行分析。设计时应确保正向和逆向扩散均具有较好的均衡性。

低计算复杂度：采用异或（XOR）、循环移位（Rotate）、模加（Modulo Addition）等低成本运算，以减少计算资源的消耗，确保适用于资源受限的环境，如物联网和嵌入式设备。

并行计算能力：设计应支持并行计算，使算法在硬件和软件实现时都能提高运算速度，确保线性变换能够并行处理多个字节块，以提高吞吐率。

抗代数攻击：避免构造简单的仿射变换，以增加密码分析难度，使攻击者难以通过代数分析手段破解算法。

**3.3 非线性层设计**

S 盒（S-box）是密码算法中的核心非线性组件，其安全性直接影响整个密码系统对抗攻击的能力。为了确保 S 盒能够有效抵抗差分攻击、线性攻击和代数攻击，设计时需要满足平衡性、完全性、高代数次数、低最大差分概率、低最大线性概率以及无不动点等基本要求。然而，在实际设计过程中，还需要进一步优化 S 盒的结构，使其在计算复杂度和安全性之间取得良好平衡。除了前述的基本特性外，S 盒的设计还应考虑以下安全性因素：

平衡性：S 盒必须是一一映射，即输入字节和输出字节应均匀分布，以确保所有输入具有相同的概率映射到所有可能的输出值，避免出现偏差。

完全性：任何输入字节的变化都应影响所有输出字节，确保密码结构的高度复杂性，使得攻击者无法通过部分字节的变化推测出其他字节的信息。

高代数次数：S 盒的代数次数越高，抗代数攻击能力越强。通常，代数次数应尽可能接近最大可能值，以防止攻击者利用代数关系求解 S 盒的映射。

低最大差分概率：S 盒的最大差分概率（Maximum Differential Probability）应尽可能小，以增强抗差分攻击能力。

低最大线性概率：S 盒的最大线性概率（Maximum Linear Probability）应尽可能小，以降低被线性攻击破解的可能性。

无不动点：S 盒不应存在不动点，即输入等于输出的情况（），以防止攻击者利用固定点构造攻击路径。

**3.4 JH结构抗攻击能力**

JH采用基于比特的固定置换的压缩函数，并适用MD结构的域扩展方式以及特定的填充规则构建杂凑函数。对于JH结构的不可区分性，假设底层置换是一个比特的随机置换的前提下，JH结构在输出长度为比特的情况下，与随机预言机的不可区分性又是界限为

其中，为区分器查询的总次数，为区分器查询的总数据块数。JH 所使用的填充规则至关重要：若不采用长度填充，则对输出为 比特的 JH 结构，存在一个查询复杂度为常数的简单区分攻击，能够突破其不可区分性。

对于JH压缩函数的抗碰撞能力，在目前公开文献中证明：对于，若使用理想的-比特置换并通过输出-比特的杂凑函数，则JH杂凑函数可抵抗次查询的碰撞攻击。该结果表明，JH 杂凑函数在随机置换模型下达到了抗碰撞攻击的最优理论界限。

4.一种面向字节的杂凑密码设计

本项目提出了两种杂凑密码设计方案，采用了不同的密码结构和轮函数的设计，目标是提升在安全性分析和性能上的表现。这两种设计方案分别基于4比特或5比特大小的S盒，另一个则采用8比特的S盒；线性变换方面，一个采用整体状态的置换，另一个则使用4×4的最大差分集（MDS）矩阵，其中每个元素是一个字节。这些设计的差异会直接影响密码的安全性和性能，因此在具体的应用场景中，设计者可以根据需求来选择最适合的方案。杂凑密码广泛应用于物联网、区块链等先进技术领域，而随着这些应用的不断发展，能够灵活调整杂凑密码参数以满足不同需求变得尤为重要。然而，人工维护每个参数会造成巨大的资源和时间成本，每次修改后还需要重新评估安全性和性能，显然这种方法不仅低效且容易出错。因此，本项目提出将这些步骤模块化，从而极大地提升杂凑密码设计的效率，降低人工干预，扩展其应用前景。模块化设计不仅简化了设计流程，还可以为不同的应用场合提供定制化的解决方案。

为了更好地实现杂凑密码的自动化设计，本项目提出了几个关键性的设计准则，确保设计的密码既具备强大的安全性，又能够满足不同应用环境下的性能需求。在非线性层的设计方面，S盒是最重要的组件，因此需要选择符合严格标准的S盒。S盒的设计标准包括其必须是可逆的且没有固定点，这能有效防止线性化攻击，同时要求每个输出比特至少依赖于4个输入比特，以增加密码的复杂性。此外，S盒的代数次数至少为2，这保证了密码系统在面对代数攻击时的抗性。同时，为了抵抗差分攻击和线性分析，S盒的差分概率和线性偏差需要尽量为1/4，差分和线性分支数要尽量达到3。这些设计要求使得S盒具有极强的抗攻击能力，能够有效避免潜在的安全漏洞。在线性层的设计上，异或和循环移位操作的组合是基础。通过循环移位操作，设计者可以确保每个状态字之间有较好的扩散效果，以增强密码的安全性。循环移位操作的参数选取也十分重要，必须确保其分支数至少为3，防止差分和线性分析中的简化攻击。此外，设计中还要求使用不同的循环移位参数，以保证每个字之间的扩散效果良好。为了平衡性能与安全性，部分移位参数可以设为零，从而提高性能，同时保持一定的安全性要求。在实际设计中，通过自动化技术搜索符合这些条件的循环移位参数，能够有效提高设计效率并保证密码的强度。

**4.1 一种面向字节的杂凑密码设计框架**

本设计方案围绕着自动化设计一种面向字节的杂凑密码展开，旨在构建一种能够在安全性分析和性能优化上表现卓越的杂凑算法，以满足当前计算环境中对高效性和强安全性的双重需求。为实现这一目标，我们提出了一种自动化设计程序，能够通过用户输入的具体参数，自动生成面向硬件实现的杂凑密码，同时确保其在软件执行中的优异性能。该方案设计的杂凑算法支持512比特的摘要长度，以适应现代应用对数据完整性验证的需求。为了满足安全性要求并有效避免可能的碰撞和原像攻击，杂凑函数的状态大小被设置为至少1024比特，这一设置使得系统能够在处理大规模数据时依旧保持较高的安全性和稳定性。考虑到硬件实现的需求和算法的并行计算效率，杂凑密码的状态结构被设计为一个4×4×8的立方体模型，其中每一个小立方体代表一个8比特的字节。

在实际应用中，用户首先根据需求输入一系列关键参数，包括输入消息的长度、初始状态的大小、算法的轮数、输出摘要的长度等。这些参数将作为输入，指导系统进行结构生成。根据用户的输入，系统会自动选择和配置所需的各类组件，包括非线性组件（如S盒变换）和线性组件（如扩散层结构），从而确保杂凑函数的混淆与扩散特性满足安全性标准。

设计过程中，系统还会对所生成的杂凑密码进行全面的安全性评估，评估内容包括对抗碰撞攻击、差分攻击、线性攻击、第二原像攻击等常见攻击类型的抵抗能力。通过数学推导和实验验证，确保算法在不同应用场景下满足多层次的安全需求。在性能方面，系统通过模拟与实际硬件平台（如FPGA或ASIC）上的部署，评估算法的吞吐率、资源占用、功耗等关键性能指标，确保在高效性和资源消耗之间取得最佳平衡。最终，设计方案会输出完整的杂凑密码实现代码，这些代码涵盖了硬件描述语言（HDL）代码，以适应FPGA和ASIC等硬件平台的部署，同时还提供适用于软件平台（如C、Python、Rust等）的实现代码。

该设计方案采用模块化架构，每个设计步骤均以独立模块实现，用户可以根据需求灵活调用这些模块并调整相关参数。每个模块都配有详细的调用接口和参数说明，确保用户能够根据具体应用场景灵活定制杂凑算法。如图4.1所示，展示了设计的整体框架。

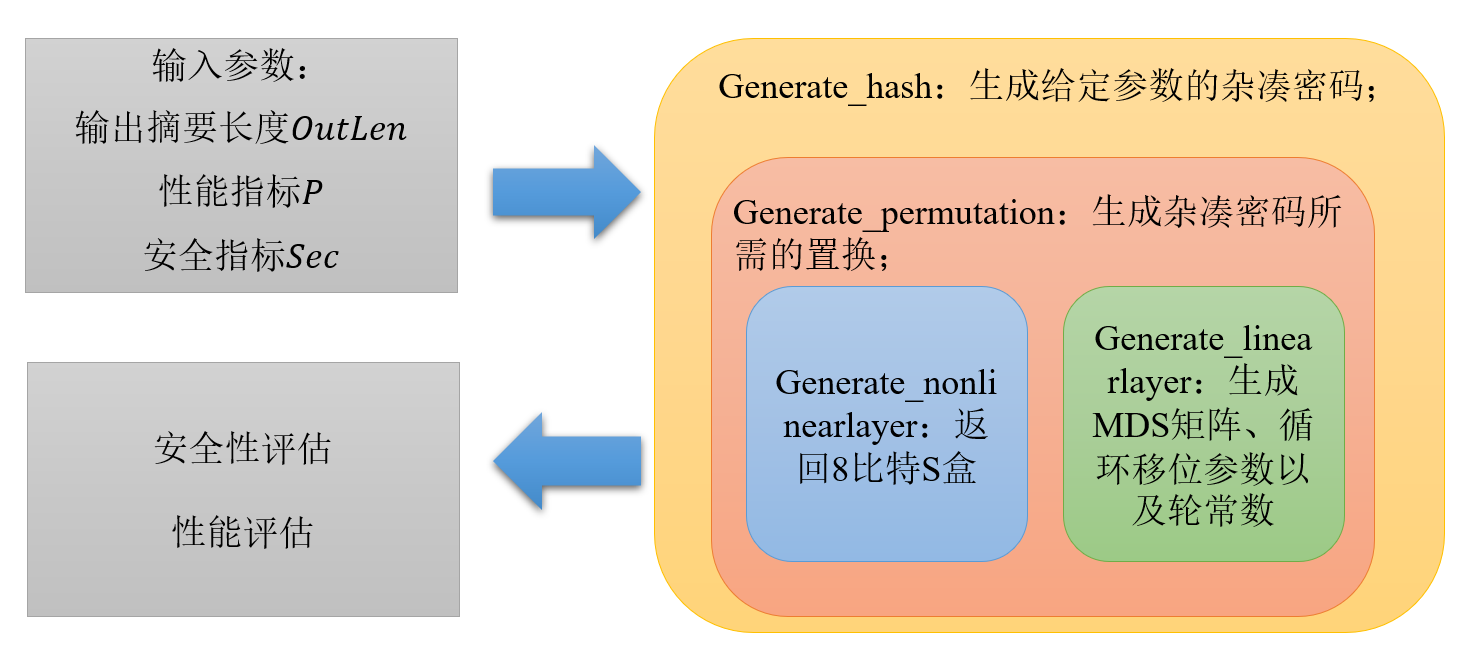


图4.1 面向字节的杂凑密码设计整体框架

**4.2 整体结构和压缩函数**

在本杂凑函数的设计方案中，整体结构遵循经典的迭代式构造模式，即由压缩函数反复迭代形成最终的杂凑值。首先，对于原始消息，系统将其按照标准化格式进行填充（padding），使填充后的消息长度为 512 比特的整数倍。填充后的消息被划分为若干个等长的消息块，记作 ，其中 ，表示总共划分为个块。初始化向量 是系统设定的一个初始状态，其长度为 比特，显著大于单个消息块长度。这种设计保证了算法在迭代过程中具备更高的信息携带容量和扩散能力，从而提升安全性。每轮迭代通过压缩函数 实现当前状态的更新，状态更新规则定义为：

在此结构中，压缩函数的功能是将一个长度为 1024 比特的中间状态值与一个长度为 512 比特的消息块共同映射为一个新的 1024 比特中间状态值。因此，该压缩函数实质上是一个 的双射映射。

在处理完所有消息块后，最终的中间状态 将被输入到输出变换函数 中，以生成最终的杂凑值。输出变换记为，因此。输出变换定义为：。其中是一个截断操作，表示将中的前比特保留，其余部分丢弃。需要注意的是，由于使用置换函数作为压缩函数，所以。压缩函数是一个双射函数，其中E是一个定义在 1024 比特输入空间上的置换函数，用于在每轮中提供非线性变换与良好的扩散特性，如图4.2所示，

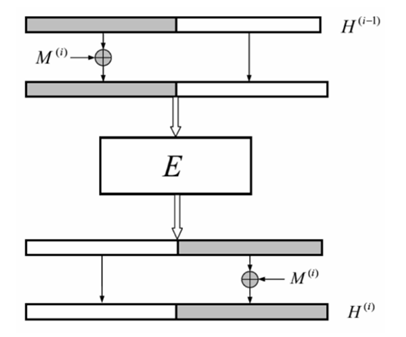


图4.2 压缩函数

定义为：

为支持上述设计，自动化系统中定义了核心函数 Generate\_hash，其主要功能是根据给定的参数生成杂凑密码的实现代码及其各个构造模块。该函数的输入参数包括： Len\_hash, 期望生成的杂凑值长度，单位为比特。该函数的返回值为：一个根据输入参数自动构造出的杂凑密码，包括压缩函数、状态更新逻辑、输出变换逻辑等核心部分的具体实现。

在 Generate\_hash 函数内部，系统调用若干子模块以完成结构构建。其中，关键子模块 Generate\_permutation 用于自动生成用于压缩函数中置换操作的函数E。此模块通过混合若干基础密码构造（如分组置换、字节级旋转、S盒替代、线性扩散层等），生成具备高非线性度、高分支数和良好代数复杂度的置换函数，以确保整个杂凑结构具备良好的抗攻击能力和可实现性。

**4.3 置换函数**

置换函数的状态设置为 构成的立方体，每个方块大小为 比特，总体状态大小为 比特。将整个状态记为，那么()代表其中一个字节。状态中橙色的Slice用()来表示，绿色的Row用表示，红色的Column用表示，蓝色的Lane用表示。如图4.3所示

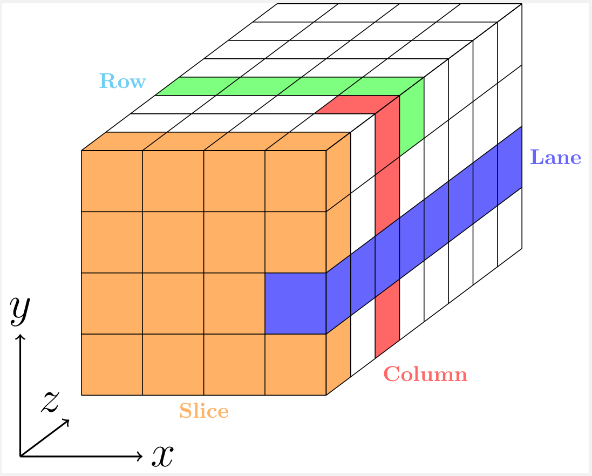


图4.3杂凑密码状态描述

置换由以下五个组件构成，分别是轮常数加、字节替代（S盒）、列移位、行混淆、Lane变换，我们将它们分别记为轮常数加：、字节替代：、列移位：、行混淆：和Lane变换：。因此输入状态经过置换得到输出定义为：

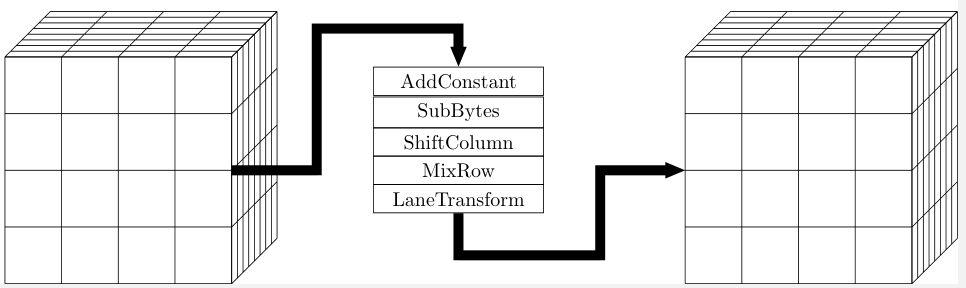
如图4.4所示：

图4.4 置换中轮函数

Generate\_permutation函数：

函数功能：用于生成杂凑密码所需的置换；

函数返回：用于生成杂凑密码所需的置换。

子模块函数：

GenerateNonlinearlay：返回目前已有的性质良好的8比特S盒；

GenerateLinearlayer：返回所需要的线性组件；

GenerateRoundconstant：返回置换中所需的轮常数。

**4.4 轮函数的线性层**

4.4.1列移位

列移位是在每一个Slice上进行的线性变换操作，旨在通过调整列的顺序来增强算法中的扩散性。其数学定义如下：

其中表示的是每一列所要循环移位的参数表示左循环移位操作。列移位的核心目的是通过在每个Slice内调整列的位置，增强密码算法在不同状态之间的扩散特性，进而提升其抗攻击能力。

为了获得最佳的扩散效果，需要精确地优化每列的移位参数。为此，采用自动化搜索技术，结合数学优化方法，如MILP（混合整数线性规划）或SAT（布尔可满足性问题）求解，进行参数优化。具体而言，目标是通过调整移位参数，使得状态的扩散度达到最大。此过程通过将扩散约束转化为优化模型，设定目标函数为最大化扩散的数量，或者通过设定最大扩散数的约束来引导搜索过程。如图4.5所示，

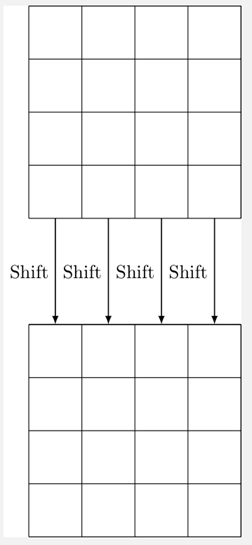


图4.5 列移位

4.4.2 行混淆

行混淆操作的设计参考了Keccak算法的思路，目的是通过对不同维度的行进行混淆，从而增强密码算法的扩散性和抗攻击性。该操作通过引入MDS（最大距离可区分）矩阵来实现，其中MDS矩阵被用来对每一行的状态进行线性变换，以保证较强的状态传播效果，进而提高算法的安全性。行混淆操作定义为

行混淆变换的主要设计目标是遵循广泛的传播策略。同样使用自动化搜索技术来搜索出使得分支数最大的MDS矩阵，将行状态之间的约束以及MDS矩阵的计算量进行约束，之后将搜索MDS矩阵转化为MILP问题或者SAT问题，设置目标函数为最大差分、线性分支数或者将最大差分、线性分支数设置为约束，接着对模型进行求解来获得最大分支数的MDS矩阵参数。

行混淆变换的设计目标是实现广泛的状态传播，确保输入比特的变化能够迅速且均匀地扩散到整个状态矩阵中，从而增强密码算法的安全性。为了优化这一过程，采用自动化搜索技术来选择最大化扩散效果的MDS矩阵。搜索过程不仅需要考虑行状态之间的传播效果，还需要对MDS矩阵的计算复杂度进行约束。通过将MDS矩阵的搜索问题转化为MILP问题或SAT，可以进一步优化设计。搜索的目标通常是最大化差分传播或线性分支数，或者将这些指标作为约束条件，进而在模型求解的过程中获得最适合的MDS矩阵参数。

4.4.3 Lane变换

在密码算法中，Lane变换操作通过对每个Lane进行循环移位来增强数据的扩散性，从而提高算法的安全性。Lane变换的定义如下：

其中表示每一个Lane的循环移位参数。移位参数同样使用类似于列移位使用自动化搜索技术获得。⋘表示左循环移位。通过这种操作，每个Lane中的元素会根据预设的移位参数进行循环移位，进而实现状态的有效扩散。Lane变换不仅增强了状态的传播速度，还增强了密码算法对差分攻击和线性攻击的抵抗力。

为了优化Lane变换的效果，移位参数​需要通过自动化搜索技术来确定。类似于列移位和行混淆操作，移位参数的优化过程通过自动化搜索实现，目的是寻找能够最大化扩散性的移位模式。通过这种方法，可以确保每个Lane的状态变换能够在算法的多个回合中产生均匀且强烈的传播效果。

GenerateLinearLayer函数：该函数的主要功能是生成所需的线性组件，以支持列移位、行混淆和Lane变换等操作。该函数返回经过设计和优化后的线性组件，从而保证整个密码算法在每一层的高效扩散性和安全性。

子模块函数：

ColumnShift：用于生成列移位所需的参数，确保在每个Slice中进行合适的列移位，以最大化状态的扩散。

RowMix：用于生成行混淆所需的MDS矩阵，确保行之间的状态能够广泛传播，从而提升密码算法的抗攻击能力。

LaneTransformation：用于生成Lane变换所需的循环移位参数，通过精确优化这些参数，进一步提高数据在Lane之间的传播效率。

**4.5 轮函数的非线性层**

SB（Substitution Box）变换是杂凑函数中的唯一非线性变换，主要用于增加算法的非线性复杂度，从而增强算法的抗攻击能力。SB变换的数学定义如下：

这里，S表示8位的S盒映射函数，应用于输入值的每个比特块。SB变换的目标是通过非线性映射打乱输入比特之间的关系，从而使得输出与输入之间的依赖关系变得更加复杂，防止简单的线性分析。如图4.21所示，

选择8位S盒作为SB变换的核心变换，其优势主要体现在以下几个方面：

实现简便性：8位S盒在许多平台上提供了一个合理的权衡，既能保持较小的字长以方便硬件实现，又能提供足够的复杂性以提升密码分析的难度。在现代流行平台上，8位S盒能够高效实现，且其大小与计算性能之间具有良好的平衡。由于8位S盒的大小适中，它允许选择!种不同的置换，这使得其组合具有较高的复杂性，有效提升了安全性。

单一S盒的使用：选择一个单一的S盒而非多个不同的S盒，有助于简化硬件实现。在密码算法中，使用多个S盒可能会增加实现复杂性并影响性能。相比之下，使用单一的S盒可以在实现上获得更高的效率，尤其是在硬件设计中，这种简化有助于减少硬件资源的消耗，并提高处理速度。

结构化S盒的优势：相较于随机生成的S盒，结构化的S盒在设计时考虑了密码学安全性和硬件实现效率。因此，结构化S盒能够在硬件上更高效地实现。其固定的结构特性使得其在硬件中可以通过预定的优化策略进行实现，从而提升整体性能。结构化S盒的设计原则基于已知的密码学理论和分析结果，因此在安全性和硬件效率方面得到了广泛验证和深入研究。

基于已有的高质量S盒：选用的S盒通常来自于已有的经过深入研究的S盒集合，这些S盒在密码学上表现出优良的抗攻击特性，尤其是对差分攻击和线性攻击具有较强的抵抗力。因为这些S盒经过广泛的测试和分析，确保它们具备较高的安全性，同时在硬件和软件中也有良好的实现性能。

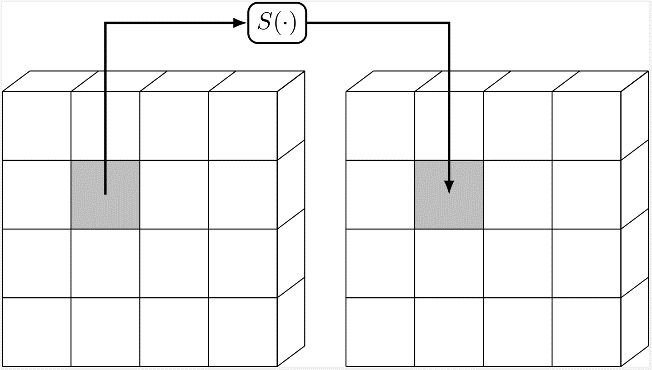


图4.6 非线性变换

GenerateNonlinearLayer函数：用于生成杂凑函数中唯一的非线性变换模块，即S盒层（Substitution Layer），其核心任务是选择并返回一个目前已知在密码学性能和实现效率方面均表现优异的8位S盒。该S盒用于构建SB变换，从而引入强非线性，打破输入比特之间的线性关系，显著提升密码算法对差分分析、线性分析等经典攻击方式的抵抗能力。

函数返回：函数返回一个固定的、结构化的、8位输入/输出的S盒，该S盒具备良好的密码学性质，例如高非线性度、低差分均衡性（差分均匀性低）、良好的代数复杂性等。同时，该S盒已在多个标准密码算法中得到验证，在硬件与软件环境下均具备高效实现能力。

通过GenerateNonlinearLayer函数生成的非线性组件，是整个密码构造中确保混淆特性（confusion）的关键部分，配合线性层的扩散特性（diffusion），共同构建起抵御多种攻击手段的坚固基础。

**4.6 安全性分析建模**

在设计杂凑密码算法时，置换的安全性至关重要，需要能够有效抵抗差分分析和线性分析等经典密码攻击。同时，整个杂凑密码算法还需具备抗碰撞分析、抗原像分析和抗第二原像分析的能力，以确保其在实际应用中的安全性和可靠性。为了评估所设计的杂凑密码算法是否符合安全标准，需要对生成的密码算法进行深入分析。

差分分析是传统且影响较大的攻击方法，主要通过研究输入的微小变化如何影响输出的模式来推导密码算法的性质。考虑到杂凑密码的结构特点及其置换操作的特性，我们选择基于MILP自动化建模的分析方法，以更精准地刻画算法的安全性。针对差分分析的防御能力，核心目标是寻找杂凑算法置换的差分特征，并确定在这些特征达到随机置换无法区分的水平时所需的最小轮数。这一下界的计算为密码设计提供理论依据。

抗碰撞分析和抗原像分析是杂凑函数安全性评估的核心内容。抗碰撞性要求攻击者很难找到两个不同的输入产生相同的杂凑值，而原像分析关注的是在给定杂凑值的情况下能否逆推出原始输入，抗第二原像分析则是确保在已知输入的情况下，难以找到另一个不同的输入映射到相同的杂凑值。为了评估这些安全性指标，需要计算杂凑算法在可行计算范围内找到碰撞或原像的计算复杂度下界，以判断杂凑算法的安全性是否足够可靠。

本项目采用 MILP（混合整数线性规划）等自动化建模技术来进行杂凑密码的安全性分析。基于MILP的整个分析流程包括解析生成的杂凑密码代码，提取关键运算模块，如置换、S 盒、异或扩散等，并构建数学模型，将密码算法的操作映射为 MILP 约束条件。随后，基于输入的安全性评估参数，例如目标轮数、差分特性、碰撞复杂度等，调用求解器计算最小安全轮数，并确定密码算法对不同攻击手段的安全性边界。最终，分析结果用于优化杂凑算法参数，使其在安全性和性能之间取得平衡。

**5 面向字节的杂凑密码设计关键技术**

本项目在设计轻量级MDS矩阵用于杂凑密码线性层的过程中，经过理论分析与实验评估后认为，从循环矩阵（circulant matrix）中构造MDS矩阵是一种具有实践可行性与理论合理性的选择。这一策略之所以优越，首先在于循环矩阵结构本身的代数特性使其更容易满足MDS性质。与随机方阵相比，循环矩阵在特定参数范围内，其结构对称性与元素重用能力使得MDS矩阵在搜索空间中出现的概率更高，减少了构造中的试验与计算成本。此外，循环矩阵具有天然的构造压缩优势：设矩阵阶数为 k，在循环结构下仅需定义其首行 k 个元素即可唯一确定整个矩阵。这一特性不仅降低了参数存储开销，还使得实现过程中具备良好的资源重用性与存储一致性。

更为重要的是，循环矩阵允许在首行元素中出现多个重复项，即在保持MDS性质的前提下，可以仅使用少数几个低复杂度的非零元素。这种条目的重复性显著降低了有限域乘法的硬件代价，因为重复的轻量级常数（如1、2、3等）便于通过简单的移位和异或操作实现，从而在硬件中可减少查找表大小、降低逻辑门数量、压缩面积与功耗，相较于必须使用 k 个互异元素的Hadamard或Cauchy矩阵而言实现成本更低，效率更高。

与此同时，循环矩阵的结构特性也天然适用于支持轮函数架构与序列化实现。在迭代杂凑结构中，每轮运算常涉及状态的矩阵乘法扩散操作，使用循环结构可使该过程具备高度的可预测性与并行处理潜力，尤其在硬件流水线与资源共享设计中能够实现更高的吞吐效率和更低的延迟表现。通过对循环矩阵的适当构造与筛选，不仅可以有效地实现强扩散性要求（即高分支数），同时还可将其嵌入到轻量级、高效能的密码实现方案中。

因此，从MDS矩阵的存在性、构造代价、实现复杂度以及与实际密码结构的兼容性多个角度综合考虑，本项目认为，优先在循环矩阵空间中搜索轻量级MDS矩阵是一种兼具理论保障与工程价值的设计策略。这一方法为构建高效、可实施、安全边界明确的杂凑密码提供了基础性的线性扩散组件，也为轻量级密码学的发展提供了新的构造视角。

**5.1 MDS矩阵构造**

衡量矩阵是否对乘法计算高效的指标是矩阵的异或运算总个数，这是一个易于理解和计算的效率指标，实践中MDS矩阵经常选择降低异或和来简化矩阵乘法计算，从而降低硬件面积，提高计算效率。值得注意的是，这个指标容易和 Hamming 重量相混淆。虽然低 Hamming 重量在实现中通常需要更少的硬件资源。例如，AES的 MDS矩阵的系数被设置成1、2、3。然而，一些研究表明，虽然这个启发式方法通常是正确的，但并不总是适用。由于一些减少效应，并且根据定义计算有限域的不可约多项式，一些不那么低 Hamming 重量的矩阵也可能用只有非常少的异或运算总个数。

本项目找到了使用公式计算循环矩阵的异或运算总个数的方法:

首先，定义单个元素的异或运算个数：对某个有限域GF()上的元素a，a的异或运算个数就是实现a和任意GF()上的元素b相乘所需要的异或运算的数量。

然后，我们给出矩阵M中某一行异或运算总个数N的计算公式:

其中是M中那一行的第个元素的异或运算个数，是那一行中非零元素的个数，而r是有限域的维度。根据杂凑算法的设计，这个有限域是GF()，所以r = 8。

由于本项目中计算的矩阵都是循环矩阵，显然M的异或运算总个数为4N。

接下来需要计算，也就是单个元素的异或运算个数。这里存在一个问题：有限域上的某个元素对于模不同的不可约多项式乘法，具有不同的异或运算个数。有趣的是，可以证明：虽然有限域上每个元素模不同的多项式时具有不同的异或运算个数，但一个有限域上所有元素的异或运算个数和是一个定值，与不可约多项式无关。计算某个有限域GF()下所有元素异或运算个数和的公式如下：

虽然如此，这个结论并不代表不同的不可约多项式对于矩阵乘法的计算效率没有影响。我们在研究过程中发现：因为最后被选取的轻量级矩阵具有的异或运算总个数较少，它们每一个元素具有的异或运算个数也较少。所以，在选取不可约多项式的时候，比起使得所有元素中异或运算个数分布较为均匀的不可约多项式，那些使得异或运算个数分布较不均匀的不可约多项式是更优的选择，因为它们的异或运算个数更多的分布于一些特定的元素上，而总的来说，有限域的阶数要远大于矩阵的阶数，只有一小部分元素会出现在矩阵当中，则那些异或运算个数较大的元素在矩阵中更不可能出现。这样在所有元素总的异或运算个数相同的情况下，标准差较大的不可约多项式下更有可能找到更轻量级的矩阵。这项发现使得我们能够在搜索时优先使用最好的不可约多项式，节约了搜索时间。为此，我们计算了GF()上所有不可约多项式下每个元素异或运算个数的标准差并制成了以下三张表格：

GF() 上各不可约多项式下每个元素异或运算个数的标准差（1）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| x | 0x165 | 0x18B | 0x163 | 0x11B | 0x13F | 0x15F |
| σ | 6.868 | 7.362 | 6.415 | 7.530 | 5.577 | 5.058 |

GF() 上各不可约多项式下每个元素异或运算个数的标准差（2）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| x | 0x1C3 | 0x139 | 0x11D | 0x177 | 0x1F3 | 0x169 |
| σ | 7.463 | 6.758 | 7.396 | 5.300 | 5.754 | 6.757 |

GF() 上各不可约多项式下每个元素异或运算个数的标准差（3）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | 0x1BD | 0x1E7 | 0x12B | 0x1D7 |
| σ | 5.253 | 5.884 | 6.175 | 5.798 |

随后，我们计算了GF()上各个不可约多项式对0-255每个元素进行乘法时所需要的异或运算个数并制成异或运算个数表。通过这个异或运算总个数表，就能够计算得到给定矩阵的异或运算总个数，从而筛选出具有较小异或运算总个数的矩阵，也就是在GF()下进行矩阵乘法的计算开销更少的矩阵。

在本项目中，我们致力于从循环矩阵（Circulant Matrix）中高效搜索MDS矩阵，以用于构建轻量级且高效的加密算法结构。为实现该目标，我们提出了一种以“轻量级性优先”为策略的候选矩阵筛选与验证方法。然而，在执行该方法过程中，我们面临两个关键性的挑战。

首先，对于阶数为 k 的一般循环矩阵（在不考虑矩阵条目具体取值的前提下），其结构可表示为 ，即首行确定后，其余各行通过首行的循环右移获得。在此设定下，首行中 k 个元素的排列方式共有 k! 种。显而易见，随着矩阵阶数 k 的增大，该组合数量呈阶乘级别增长，导致搜索空间迅速膨胀，进而使穷举式的全排列遍历方法在计算上变得难以承受。

其次，在选取用于构造矩阵的 k 个非零条目时，条目之间并不要求互异，即允许存在重复项。这种构造自由度虽提升了矩阵设计的灵活性，但同时也带来了额外的复杂性。与只考虑互异条目的排列相比，允许重复项的选择策略使得搜索空间进一步扩大，甚至可能达到指数级的增长。这使得直接对所有可能配置进行遍历与验证在实践中变得更加不可行。

因此，如何在保证构造矩阵轻量级性的前提下，有效压缩搜索空间，并准确识别其中具有 MDS 性质的结构，成为本研究需重点突破的难点问题。

为有效压缩循环矩阵的搜索空间，本项目引入了一种基于置换操作的等价关系，用以对大量候选矩阵进行归类。若存在两个置换矩阵 P 与 Q，使得对于任意两个矩阵 M 和 M'，满足 M' = PMQ，则称 M 与 M' 之间存在置换等价关系，记作 。在该等价关系的框架下，循环矩阵的集合可被划分为若干个置换等价类，每个等价类包含所有与某个循环矩阵在置换意义下等价的其他矩阵。

这种等价划分具有重要的理论与实践意义。首先，由于同一等价类中的所有循环矩阵在结构上可通过行列置换相互变换，它们在密码应用中通常表现出一致的扩散性能，例如具有相同的分支数（branch number）。因此，若目标是在所有结构上不同的循环矩阵中筛选出满足 MDS 性质的代表，我们仅需对每个等价类选择一个代表元素进行检验，便可避免大量冗余的重复计算，从而大幅提升整体搜索效率。

本研究进一步推导并证明，对于阶数为 k 的循环矩阵，其在置换等价关系下所构成的等价类总数为：。我们在此基础上对循环结构进行了系统性分析。研究表明，在阶数 的情形下，所有可能的 MDS 循环矩阵可被归纳为五种基本类型。这一分类依据矩阵第一行中不同条目的数量进行划分，具体如下：

* 类型 1：无重复条目；
* 类型 2：包含一对重复条目；
* 类型 3：包含两对重复条目；
* 类型 4：包含三对重复条目；
* 类型 5：存在三个重复条目。

上述分类方法不仅在理论上具有清晰的结构划分依据，同时也为后续的矩阵生成与验证策略提供了明确的搜索框架。通过分别枚举每种类型中重复元素的个数与位置，可系统生成所有结构可能性，从而对其是否满足 MDS 性质进行高效检验。

下表列举了从3-8阶下循环MDS矩阵可能属于的类型：

**3-8阶下循环MDS矩阵的可能类型**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 阶数 | 没有重复条目 | 1对重复条目 | 2对重复条目 | 3对重复条目 | 3个重复条目 |
| 3 | ✓ | ✓ |  |  |  |
| 4 | ✓ | ✓ |  |  |  |
| 5 | ✓ | ✓ | ✓ |  |  |
| 6 | ✓ | ✓ | ✓ |  |  |
| 7 | ✓ | ✓ | ✓ | ✓ | ✓ |
| 8 | ✓ | ✓ | ✓ | ✓ | ✓ |

在完成等价类划分与矩阵类型归类之后，便可基于上述筛选策略，对选定的轻量级循环矩阵集合进行快速且系统性的筛查，以识别其中是否存在满足MDS性质的矩阵。在具体实现过程中，本项目选取有限域 作为构造与验证矩阵性质的运算基础，并采用模多项式，即十六进制表示的 0x11B，作为该域上的最优本原多项式。在此基础上，本项目对前述筛选出的候选轻量级循环矩阵逐一进行MDS性质验证，具体包括计算其生成的线性变换的分支数（branch number）、最小距离、可逆性等核心指标。通过该方法，最终从海量候选结构中成功筛选出10个满足MDS条件的轻量级循环矩阵，用于后续杂凑函数的设计。

**5.2 大状态算法分析结果**

借助现有的自动化分析技术可以更好的实现差分分析和线性分析，但是受限于算力，自动化分析技术在面对大状态的表现不尽如人意。一方面，实现差分分析和线性分析需要构建S盒的差分分析表和线性逼近表，而大比特S盒的差分分布表和线性逼近表的模型难以精简地构建；另一方面，输入状态长度较长时，模型将膨胀到难以在可接受时间内求解的地步，这进一步导致模型求解困难，所以需要一种有效的优化方案来使模型能够给出更好的差分分析和线性分析结果。首先需要约束输入差分（掩码）和输出差分（掩码）的汉明重量，其次需要将前面得到的概率（偏差）作为反馈，来减少搜索的路径数量，最后，利用循环移位的特性，尽可能让差分（掩码）中的非零部分相互抵消。如果上述优化方案都难以让模型在一定时间内求解出更好的结果，将选择一种基于深度优先遍历的算法来搜索出更好的结果。该算法主要是将搜索过程看作遍历树中的每一个节点，为了尽可能少的访问节点，需要通过剪枝法将一些已经不满足条件的节点从树中剪掉，从而让算法不再访问子树的剩余节点，这样可以加速算法的运行速度。

由于面向字的杂凑密码使用8比特的S盒作为其非线性层，当使用自动化技术搜索比特级别的差分或线性路径时，S盒的差分分布表或者线性近似表都需要大量的约束来描述，并且由于状态较大，搜索模型常常难以在有效的时间内给出解。在进行Rebound分析时，需搜索到合适的（相关密钥）截断差分路径，再结合使用状态和密钥的自由度来得到（自由/半自由）起始碰撞分析。

6. 新设计的杂凑算法描述及其安全性分析和实现

**6.1 杂凑算法描述**

6.1.1 符号与约定

所有数据变量用最高比特（或子块）在左边，最低比特（或子块）在右边表示，而且从左到右从0开始编号。例如，一个n比特数从左到右分别称为它的第0比特、第1比特、第2比特、…、第n-1比特。当一个变量被分成几个子块时，最左边（最高）子块由原始数据的最高部分组成。

当杂凑算法的输入（或输出）是字节流时，按地址小的是高位（即左边），地址大的是低位（即右边）的顺序把每4（或2）个字节组成一个32（或16）比特字。32（或16）比特字之间的顺序是先左后右，即存储时，左边地址小，右边地址大。

下面举例说明。根据以上规定，认为二进制表示的64比特数：

0000 0001 0010 0011 0100 0101 0110 0111

1000 1001 1010 1011 1100 1101 1110 1111

与如下4种16进制表示的64比特数互相等价：

0x0123456789abcdef

0x01234567||0x89abcdef

0x0123||0x4567||0x89ab||0xcdef

0x01||0x23||0x45||0x67||0x89||0xab||0xcd||0xef

运算符如下表所示。

表6.1 运算符

|  |  |
| --- | --- |
| 运算符 | 描述 |
| 0x | 用前缀0x表示16进制数 |
| = | 赋值，也可表示相等 |
| \* | 表示矩阵乘法 |
| i mod n | 表示整数i除以n的余数，0≤ i mod n≤n-1 |
| || | 字符串连接 |
| |S| | 表示字符串S的长度 |
| ◦ | 变换复合，即对于变换f1、f2和变量x：f2◦f1(x)= f2(f1(x)) |
|  | 操作数循环左移n比特 |
|  | 第i个消息块 |
|  | 第i次迭代之后算法的内部状态 |
|  | 算法内部状态A里位于(x,y,z)的元素。 |
|  | 一个字节的异或运算 |
| AB | 将运算A复合到运算B上 |
|  | 使内部状态 x 经过 S 盒S |
|  | 置换函数 |

6.1.2 杂凑算法编制描述

该算法的杂凑值长度为512比特。新设计的杂凑算法推荐轮数为14轮，其选取的主要依据为杂凑算法的分析轮数。

该算法的结构如图2-1所示。​在压缩函数中，比特的输入与m比特的消息块被压缩成比特的。这样的过程重复进行次，也即轮数为。​消息摘要的大小为比特。该算法各项参数的取值如下表所示：

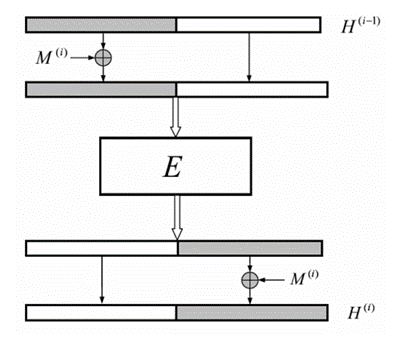


图 6.1 杂凑密码结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | n | S | m | r |
| bytehash-512 | 512 | 1024 | 512 | 14 |

消息填充。在消息M后面联接1位的“1”，然后联接最少个数的“0”，使得填充后的消息的长度为S的倍数。即：

然后将其分割为若干个512比特的消息块, 。

初始值IV。记函数的初始状态为IV，即。。如图6.2表示为一个长宽高为4\*4\*8的三维立方体。

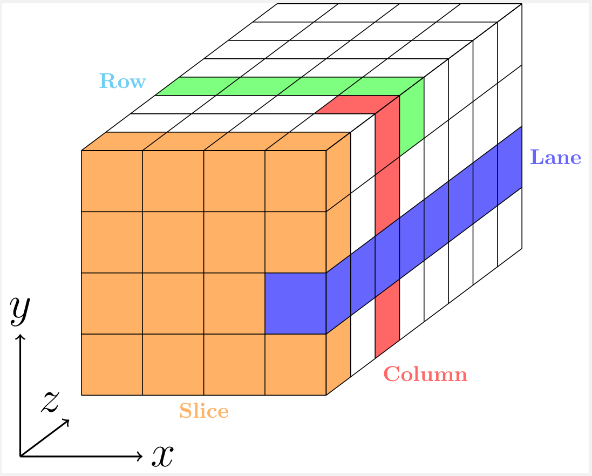


图 6.2 IV的结构

其中x, y, z坐标均从0开始。x轴和y轴张成的面记为切片（slice），每个切片根据 z 坐标取值初始化为以下对应的矩阵：

输出变换。输出变换定义如下：

。

其中是一个截断操作，表示将x中的前n位保留，其余部分丢弃。需要注意的是，由于使用置换函数作为压缩函数，所以, 其中。

置换。的置换的设计采用了分组密码的设计思想，其结构类似广义Feistel结构。记当前轮数为，，输入状态为，满足。将分为2部分，每部分64字节，即 。置换P计算流程如下：

计算;

计算;

计算;

最后，置换P返回作为第轮的输出。

轮函数。图6.3给出了轮函数的图示。

的输入是一个128字节的状态，输出是一个128字节的状态，，即，其中，分别是轮常数加(AddConstant)、字节代换(SubBytes)、列位移(ShiftColumn)、行混淆(MixRow)、道变换(LaneTransform)。

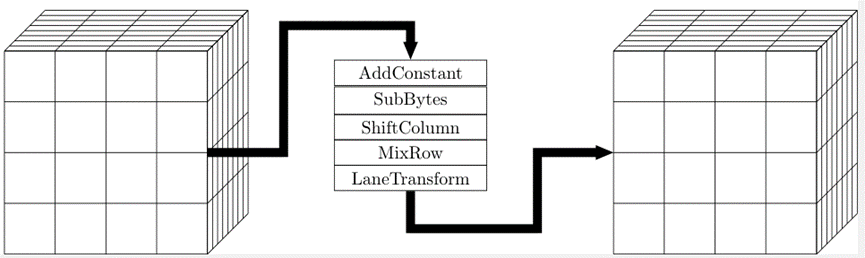


图6.3 轮函数E

轮常数加法。轮常数加在每个字上进行，定义为：

其中c表示轮常数，其表示如下：

。

字节代换。字节代换S是杂凑函数中唯一的非线性变换，定义为：

.

列位移。列移位是在每一个 Slice 上进行的，定义为：

.

其中表示的是每一列所要循环移位的参数。其表示如下：

.

行混淆。行混淆操作M定义为:

.

其中M为一个4×4的MDS矩阵。其表示如下

.

道变换。在道上做循环移位操作，定义为：

.

其中表示每一个道的循环移位参数。其表示如下：

.

S盒及其性质。本项目选择采用与 AES 密码算法中相同的 16-bit S 盒， S 盒中的每个元素是一个十六进制整数，其取值如下表所示：

表6.2 S盒

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 36 | 94 | 89 | cb | 77 | 96 | d2 | 4b | 05 | f7 | ab | c5 | 6d | a1 | d6 | 5b |
| 61 | 91 | e7 | d0 | 1f | a9 | 43 | 1d | 9b | be | f4 | b8 | 42 | 63 | 87 | bb |
| 02 | 58 | c3 | ac | e4 | e5 | eb | b3 | 83 | 70 | 64 | 20 | 57 | 08 | 60 | 85 |
| 2f | 90 | 07 | ee | 23 | 33 | 81 | 12 | 14 | ea | 39 | 21 | 62 | cd | 28 | 2e |
| 2c | f6 | dd | 25 | bc | 11 | a7 | e6 | fd | 53 | 98 | 9c | 38 | 1b | 5c | 54 |
| 75 | 95 | 26 | 00 | 09 | 3b | 44 | 9d | 15 | 5d | 1c | 9a | 5f | c9 | a4 | 78 |
| 5a | f3 | 0b | 0c | e9 | 0a | 06 | 3e | 71 | e1 | fa | f5 | 7f | 65 | 19 | df |
| 8e | 32 | fb | 74 | 50 | d9 | 72 | 24 | 45 | 0f | 69 | 76 | da | 41 | b1 | db |
| 79 | 80 | 3a | 49 | e8 | bf | 73 | 16 | 18 | 8d | ce | a3 | 0e | c6 | ef | e3 |
| d7 | 99 | 6e | 35 | fc | af | a2 | c1 | de | c2 | 1e | d1 | 6c | f1 | aa | 7e |
| 8c | 52 | d4 | 4a | 7c | 93 | f0 | e2 | d8 | 66 | 04 | 9e | 84 | 3c | 13 | ae |
| 86 | 88 | a5 | 68 | d3 | 37 | 3d | 56 | 6a | 5e | 7a | ad | c8 | b2 | 40 | 67 |
| 0d | b7 | 46 | 7d | a6 | 82 | 6b | 3f | 34 | 22 | b0 | c0 | 29 | 4e | 59 | 7b |
| c7 | 31 | ba | 47 | fe | c4 | d5 | e0 | 92 | b9 | 10 | a0 | 8b | ed | 55 | 97 |
| ca | 1a | f9 | 2a | cc | f2 | 4c | 51 | 03 | 30 | 4d | f8 | b4 | bd | cf | 48 |
| ec | 2b | 9f | ff | 27 | 17 | b6 | 8f | 8a | b5 | 01 | a8 | 6f | 4f | dc | 2d |

6.1.3 样本数据

第一组样本数据：

消息：0x00，0x00，0x00，0x00，0x00，0x00，0x00，0x00，0x00，0x00，0x00，0x00，0x00，0x00，0x00，0x00

杂凑值:

0x2077a2a38724dbe9f8b6893be8f6bd0c315fb4f76f77813dee61db0f3ca07e59a3fa6f1bf37e288ea5a1b39564f9318cb5ddf4f1461cc1bbeba69b483ff41d8f

第二组样本数据：

消息：0x00，0x01，0x02，0x03，0x04，0x05，0x06，0x07，0x08，0x09，0x0a，0x0b，0x0c，0x0d，0x0e，0x0f

杂凑值：

0x985cb2a29adea49b95fc109106e66205f45950e9d257755dd0619b8a98a08cc88d951b4e96a5fd9f729989e08af3acfc6cde482210363de99f1d44a5f528f0aa

第三组样本数据：

消息：0x00，0x00，0x00，0x00

杂凑值：

0x36452ca0803e05b8b8c15fe5578cb69daf9b92a8eab6439f9bde6f4b5519cb14c90d436865a58c18fdb3c5c4be912b885a71d03371da428ddf9b38237c6b4695

**6.2 安全性分析**

6.2.1 差分分析

本项目选取不同的lane移位参数，测试了相应置换的活跃S盒个数，具体见表6.3。根据表6.3的数据，本项目设计的杂凑算法的参数按照第11行的参数来选取。

表6.3 活跃S盒个数

|  |  |  |  |
| --- | --- | --- | --- |
|  | Lane位移参数 | r | 活跃S盒个数（合格线：171） |
| 1 | 1 1 1 1 2 2 2 2 3 3 3 3 4 4 4 4 | 6 | 75 |
| 2 | 5 1 3 5 7 3 7 0 2 4 4 2 4 5 1 1 | 6 | 75 |
| 7 | 82 |
| 3 | 1 2 1 6 2 1 4 0 6 0 6 1 5 5 3 4 | 6 | 75 |
| 7 | 82 |
| 4 | 0 2 3 7 4 0 5 6 5 5 2 3 3 7 1 2 | 6 | 75 |
| 7 | 82 |
| 5 | 7 6 3 2 3 1 6 7 2 0 1 5 1 2 4 3 | 6 | 75 |
| 7 | 82 |
| 6 | 5 7 5 5 7 6 7 1 3 4 6 6 6 5 0 7 | 6 | 75 |
| 7 | 82 |
| 7 | 3 4 2 6 0 3 1 0 6 2 5 7 1 6 7 1 | 6 | 75 |
| 7 | 82 |
| 8 | 4 5 6 7 1 6 4 6 7 3 7 1 5 0 5 4 | 6 | 75 |
| 7 | 82 |
| 9 | 5 5 7 1 4 7 0 6 1 0 6 0 2 3 4 3 | 6 | 75 |
| 7 | 82 |
| 10 | 1 0 2 0 7 2 3 2 6 1 1 7 3 5 0 3 | 6 | 75 |
| 7 | 82 |
| 11 | 2 7 2 3 0 4 3 6 4 3 1 4 1 1 5 0 | 6 | 85 |
| 7 | 128 |
| 8 | 160 |
| 12 | 0 1 0 2 6 6 5 3 1 4 6 5 4 7 1 7 | 6 | 75 |
| 7 | 82 |
| 8 | 104 |
| 13 | 1 7 1 6 7 3 6 2 2 6 2 3 4 0 7 5 | 6 | 75 |
| 7 | 82 |
| 8 | 104 |
| 14 | 5 1 1 6 1 0 4 3 0 5 6 1 3 6 2 4 | 6 | 75 |
| 7 | 82 |
| 8 | 104 |
| 15 | 7 5 7 6 2 0 3 7 1 2 4 2 5 7 6 0 | 6 | 75 |
| 7 | 82 |
| 8 | 104 |

对于本项目设计的bytehash-512算法，基于MILP自动化建模，找到了置换的活跃S盒个数，如表6.4所示。因为杂凑算法中S盒的输入输出差分成立的概率最小是2-6，而1024/6 ≈ 171，所以20轮的bytehash-512算法可以抵抗差分分析。

6.4 选取的参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列位移参数 | Lane位移参数 | Mds | r | 活跃S盒个数（合格线：171） |
| (3, 2, 1, 2) | 2 7 2 3 0 4 3 6 4 3 1 4 1 1 5 0 | 1 1 2 4 1 2 4 1 2 4 1 1 4 1 1 2 | 1 | 1 |
| 2 | 5 |
| 3 | 9 |
| 4 | 25 |
| 5 | 41 |
| 6 | 85 |
| 7 | 129 |
| 8 | 160 |
| 9 | 176 |

S盒的选取主要基于实现与密码分析的综合权衡考虑。首先，从大小的角度来看，8位 S 盒在主流平台上是最小的字长，实现方面更为简便，同时能够兼顾密码分析的安全性要求。此外，8位 S 盒的置换空间高达 种不同的可能性，为安全性提供了较大的搜索空间。其次，使用单一 S 盒而非多个不同的 S 盒，是在实现复杂度与密码分析强度之间的权衡选择，有助于降低实现成本并确保整体安全性。同时，本项目避免使用随机 S 盒，而选择了结构化 S 盒，这种选择有助于提高硬件实现的效率，尤其是在资源受限的环境中更具优势。由于该 S 盒直接继承自 ZUC 密码算法，其实现特点（特别是在硬件加速方面）已在先前研究中得到了充分验证。因此，采用 ZUC 的 S 盒不仅简化了实现过程，还确保了安全性和性能的平衡。

6.2.2 长度扩展攻击

密码杂凑算法bytehash-512在输出阶段仅输出512比特杂凑值，置换P中未知比特为()比特，长度扩展攻击的复杂度为寻找置换P中未知比特碰撞的复杂度，具体如下:

|  |  |
| --- | --- |
| 算法 | bytehash-512 |
| 杂凑长度 | 512 |
| 长度扩展攻击的复杂度 |  |

由上表可以得出,密码杂凑算法bytehash-512长度扩展攻击的复杂度不小于 。

6.2.3 Rebound攻击

正向差分路线和逆向差分路线在中间很难产生匹配，寻找两个匹配差分的复杂度大约为, 故该杂凑算法可以抵抗Rebound攻击。

活跃S盒测试结果如表6.5所示。

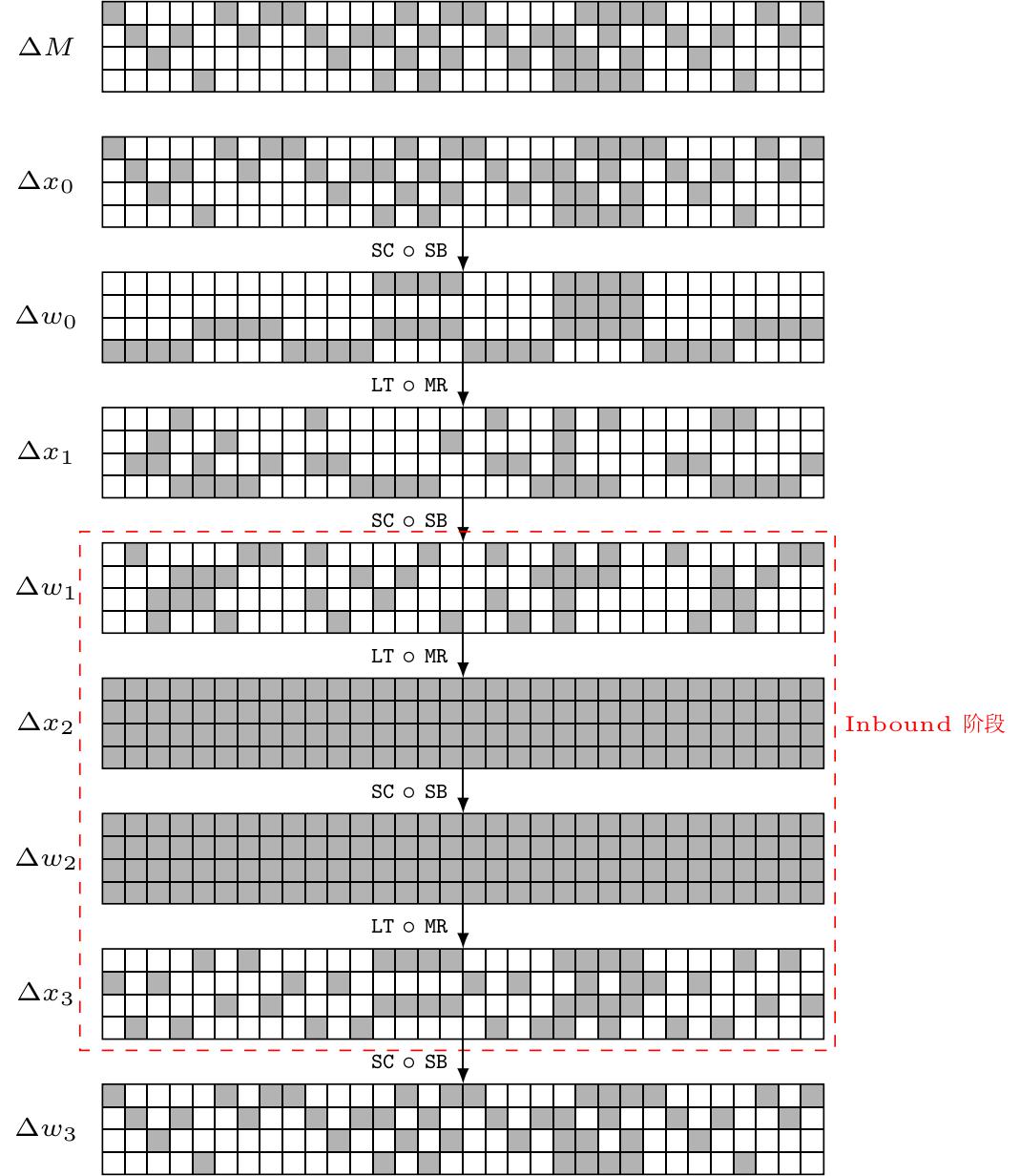
表6.5 活跃S盒测试结果

|  |  |
| --- | --- |
| 运行轮数 | 活跃S盒个数 |
| 1 | 1 |
| 2 | 5 |
| 3 | 9 |
| 4 | 25 |
| 5 | 41 |
| 6 | 85 |
| 7 | 129 |
| 8 | 160 |
| 9 | 176 |

4轮Rebound分析。

在4轮的Rebound分析中，我们利用MILP自动化方法搜索杂凑函数中置换的截断差分路线，如图所示为搜索到的4轮Rebound分析截断差分路线，其中灰色块表示差分不固定（截断差分），白色块表示差分为0。根据压缩函数可知，吸收消息时512比特的消息异或上128个字节初始值的前4比特，并且置换结束后消息需要异或128个字节初始值的后4比特。因此，搜索时输入截断差分与输出截断差分的位置需要相同。在这条截断差分路线中，Inbound阶段为

其中获取起始点的位置为的SB操作。根据差分路线可知，平均一对输入输出差分可以获得一个起始点，在Inbound阶段我们一共可以获得个起始点。而Outbound阶段的差分概率为，为与相同位置的44个字节，对于一个字节前4比特与后4比特差分相等的概率为，所以Outbound阶段的差分概率计算为。因此Inbound阶段起始点的自由度足够Outbound阶段使用，发起碰撞攻击的时间复杂度为。



Outbound概率：

图6.4 4轮Rebound分析

6轮Rebound分析

在6轮的Rebound分析中，我们利用MILP自动化方法搜索杂凑函数中置换的截断差分路线，如图所示为搜索到的6轮Rebound分析截断差分路线，其中灰色块表示差分不固定（截断差分），白色块表示差分为0。在6轮Rebound分析中，我们使用了非全活跃超级S盒技术，使得Inbound阶段可以多覆盖一轮。在这条截断差分路线中，Inbound阶段为

其中获取起始点的位置为的SB操作。根据差分路线可知，平均一对输入输出差分可以获得一个起始点，在Inbound阶段我们一共可以获得个起始点，而Outbound阶段的差分概率为，因此Inbound阶段起始点的自由度足够Outbound阶段使用，发起碰撞攻击的时间复杂度为。

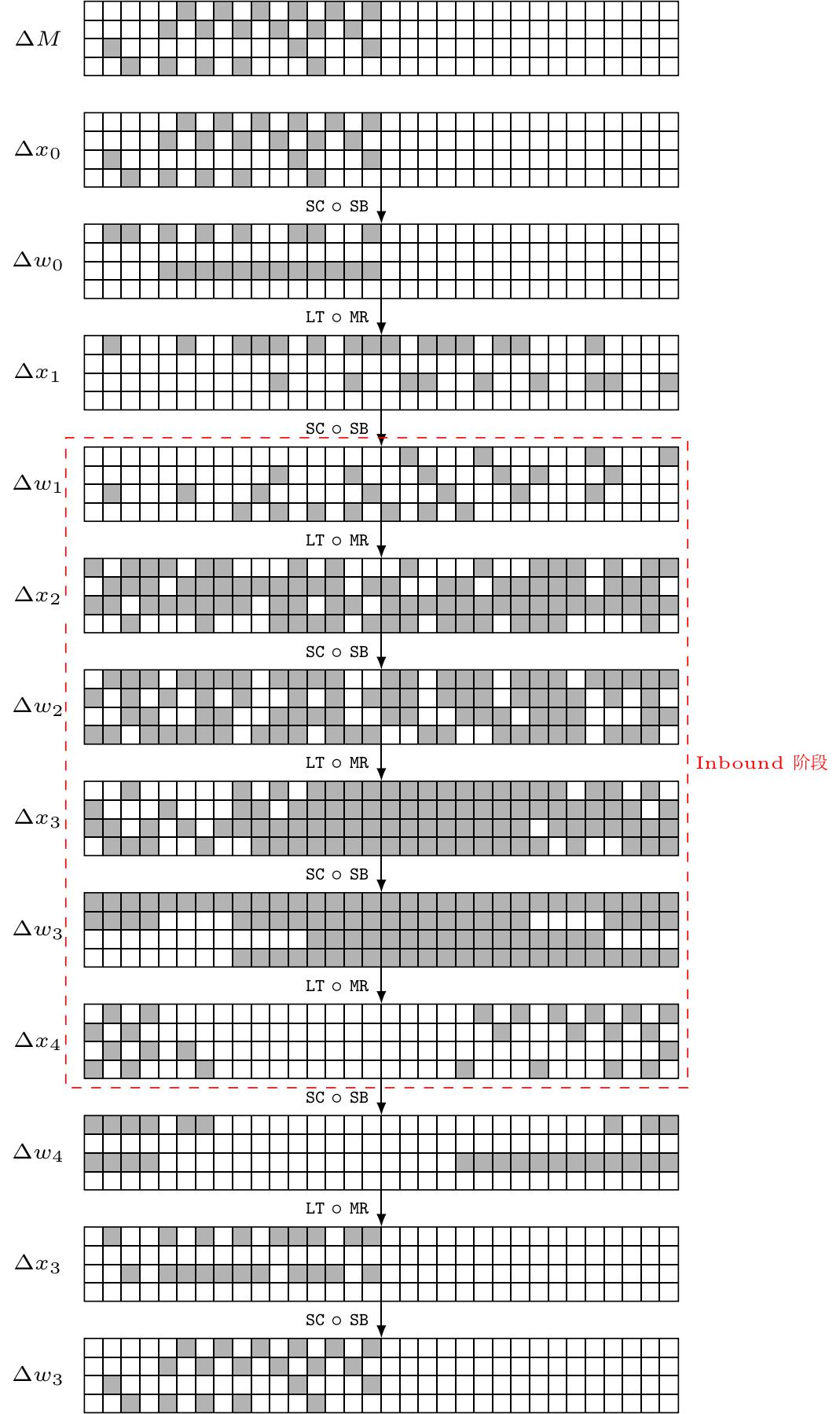


图6.5 5轮Rebound分析

Outbound概率：

Outbound概率：

**6.3 杂凑算法实现**

杂凑算法在Xilinx公司xc7a35tcpg236 - 1型号设备上，利用Vivado 2024.1进行设计、仿真与综合实现。具体说明如下。消息填充：除 Bytehash-512算法外，其余算法的实现均包含消息填充电路，且提及的消息长度均指填充前的原始长度；

算法实现：其中SHA2-512采用标准的80轮实现方式。

表6.6 杂凑算法实现

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 函数(h, r, c) | 消息长度 | 吞吐量（Gbps） | | | | 最小时钟周期（ns） | 最大轮数（以128B消息计） |
| 9轮 | 12轮 | 14轮 | 25轮 |  | |
| ByteHash-512 | / | 11.938 | 9.183 | **7.958** | 4.591 | 4.289 | 219 |
| SHA2-512 | 128B | 0.543 | | | | 11.428 | / |
| 1KB | 0.943 | | | |
| 16KB | 1.046 | | | |
| 100MB | 1.054 | | | |
| 1GB | 1.054 | | | |

7. 成果展示

本项目使用自动化分析技术，通过构建循环MDS矩阵分类模型和上每个元素的异或和个数表，实现了对MDS矩阵的快速筛选与乘法开销的精确衡量。这使得我们在生成MDS矩阵时不需要进行开销极高的逐个枚举验证，而只需要生成满足某个分类约束的的矩阵并计算出乘法开销，再对乘法开销小于要求的矩阵进行MDS性质的验证，即可得到轻量级MDS矩阵。这样设计出来的矩阵不仅具有较高的混淆能力，还能提高算法的执行效率。本项目设计的面向字节的杂密码需要用的F(28)上4x4的MDS矩阵。下图7.1展示了使用自动化技术生成部分MDS矩阵的结果。



该项目的杂凑密码是基于JH结构的，除了将自动化生成的MDS矩阵作为行混淆步骤的组件加入到杂凑密码中，还需要按照实际需求，定义轮常数、轮数和列位移。这些参数都确定了以后，即可生成完整的杂凑密码，给定明文，即可得到对应的512比特密文输出，下图7.2展示了一对输入和输出。



8. 总结展望

在面向字节的杂凑密码设计方面，项目在如何构建密码学性质良好的MDS矩阵、如何自动化建模给出新设计的杂凑算法的差分分析和Rebound分析方面取得了重要进展。这些研究为未来杂凑密码的设计与分析提供了重要的技术支持。

9. 参考文献

[1] Florian Mendel, Christian Rechberger, Martin Schläffer, and Søren S. Thomsen. The rebound attack: Cryptanalysis of reduced whirlpool and grøstl. In Orr Dunkelman, editor, Fast Software Encryption, 16th International Workshop, FSE 2009, Leuven, Belgium, February 22-25, 2009, Revised Selected Papers, volume 5665 of Lecture Notes in Computer Science, pages 260-276. Springer, 2009.

[2] Florian Mendel, Thomas Peyrin, Christian Rechberger, and Martin Schläffer. Improved cryptanalysis of the reduced grøstl compression function, ECHO permutation and AES block cipher. In Michael J. Jacobson Jr., Vincent Rijmen, and Reihaneh Safavi-Naini, editors, Selected Areas in Cryptography, 16th Annual International Workshop, SAC 2009, Calgary, Alberta, Canada, August 13-14, 2009, Revised Selected Papers, volume 5867 of Lecture Notes in Computer Science, pages 16-35. Springer, 2009.

[3] Shuang Wu, Dengguo Feng, and Wenling Wu. Cryptanalysis of the LANE hash function. In Michael J. Jacobson Jr., Vincent Rijmen, and Reihaneh Safavi-Naini, editors, Selected Areas in Cryptography, 16th Annual International Workshop, SAC 2009, Calgary, Alberta, Canada, August 13-14, 2009, Revised Selected Papers, volume 5867 of Lecture Notes in Computer Science, pages 126-140. Springer, 2009.

[4] Mario Lamberger, Florian Mendel, Christian Rechberger, Vincent Rijmen, Rijmen and Martin Schläffer. Rebound distinguishers: Results on the full whirlpool compression function. In Mitsuru Matsui, editor, Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings, volume 5912 of Lecture Notes in Computer Science, pages 126-143. Springer, 2009.

[5] Dmitry Khovratovich, Ivica Nikolic, and Christian Rechberger. Rotational rebound attacks on reduced skein. In Masayuki Abe, editor, Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings, volume 6477 of Lecture Notes in Computer Science, pages 1-19. Springer, 2010.

[6] Henri Gilbert and Thomas Peyrin. Super-sbox cryptanalysis: Improved attacks for aes-like permutations. In Seokhie Hong and Tetsu Iwata, editors, Fast Software Encryption, 17th International Workshop, FSE 2010, Seoul, Korea, February 7-10, 2010, volume 6147 of Lecture Notes in Computer Science, pages 365-383. Springer, 2010.

[7] Joan Daemen and Vincent Rijmen. Understanding two-round differentials in AES. In Roberto De Prisco and Moti Yung, editors, Security and Cryptography for Networks, 5th International Conference, SCN 2006, Maiori, Italy, September 6-8, 2006, Proceedings, volume 4116 of Lecture Notes in Computer Science, pages 78-94. Springer, 2006.

[8] Yu Sasaki, Yang Li, Lei Wang, Kazuo Sakiyama, and Kazuo Ohta. Non-full-active super-sbox analysis: Applications to ECHO and grøstl. In Masayuki Abe, editor, Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings, volume 6477 of Lecture Notes in Computer Science, pages 38-55. Springer, 2010.

[9] Yu Sasaki, Lei Wang, Shuang Wu, and Wenling Wu. Investigating fundamental security requirements on whirlpool: Improved preimage and collision attacks. In Xiaoyun Wang and Kazue Sako, editors, Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings, volume 7658 of Lecture Notes in Computer Science, pages 562-579. Springer, 2012.

[10] Vincent Rijmen, Deniz Toz, and Kerem Varici. Rebound attack on reduced-round versions of JH. In Seokhie Hong and Tetsu Iwata, editors, Fast Software Encryption, 17th International Workshop, FSE 2010, Seoul, Korea, February 7-10, 2010, Revised Selected Papers, volume 6147 of Lecture Notes in Computer Science, pages 286-303. Springer, 2010.

[11] Dmitry Khovratovich, María Naya-Plasencia, Andrea Röck, and Martin Schläffer. Cryptanalysis of Luffa v2 components. In Alex Biryukov, Guang Gong, and Douglas R. Stinson, editors, Selected Areas in Cryptography - 17th International Workshop, SAC 2010, Waterloo, Ontario, Canada, August 12-13, 2010, Revised Selected Papers, volume 6544 of Lecture Notes in Computer Science, pages 388-409. Springer, 2010.

[12] María Naya-Plasencia. How to improve rebound attacks. In Phillip Rogaway, editor, Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings, volume 6841 of Lecture Notes in Computer Science, pages 188-205. Springer, 2011.

[13] Alexandre Duc, Jian Guo, Thomas Peyrin, and Lei Wei. Unaligned rebound attack: Application to keccak. In Anne Canteaut, editor, Fast Software Encryption - 19th International Workshop, FSE 2012, Washington, DC, USA, March 19-21, 2012. Revised Selected Papers, volume 7549 of Lecture Notes in Computer Science, pages 402-421. Springer, 2012.

[14] Itai Dinur, Orr Dunkelman, Nathan Keller, and Adi Shamir. Eﬀicient dissection of composite problems, with applications to cryptanalysis, knapsacks, and combinatorial search problems. In Reihaneh Safavi-Naini and Ran Canetti, editors, Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings, volume 7417 of Lecture Notes in Computer Science, pages 719-740. Springer, 2012.

[15] Jérémy Jean, María Naya-Plasencia, and Thomas Peyrin. Improved rebound attack on the finalist grøstl. In Anne Canteaut, editor, Fast Software Encryption - 19th International Workshop, FSE 2012, Washington, DC, USA, March 19-21, 2012. Revised Selected Papers, volume 7549 of Lecture Notes in Computer Science, pages 110-126. Springer, 2012.

[16] Riham AlTawy, Aleksandar Kircanski, and Amr M. Youssef. Rebound attacks on stribog. In Hyang-Sook Lee and Dong-Guk Han, editors, Information Security and Cryptology - ICISC 2013 - 16th International Conference, Seoul, Korea, November 27-29, 2013, Revised Selected Papers, volume 8565 of Lecture Notes in Computer Science, pages 175-188. Springer, 2013.

[17] Yu Sasaki, Yuuki Tokushige, Lei Wang, Mitsugu Iwamoto, and Kazuo Ohta. An automated evaluation tool for improved rebound attack: New distinguishers and proposals of shiftbytes parameters for grøstl. In Josh Benaloh, editor, Topics in Cryptology - CT-RSA 2014 - The Cryptographer’s Track at the RSA Conference 2014, San Francisco, CA, USA, February 25-28, 2014. Proceedings, volume 8366 of Lecture Notes in Computer Science, pages 424-443. Springer, 2014.

[18] Victor Cauchois, Clément Gomez, and Reynald Lercier. Grøstl distinguishing attack: A new rebound attack of an aes-like permutation. IACR Trans. Symmetric Cryptol., 2017(3): 1-23, 2017.

[19] Xiaoyang Dong, Zhiyu Zhang, Siwei Sun, Congming Wei, Xiaoyun Wang, and Lei Hu. Automatic classical and quantum rebound attacks on aes-like hashing by exploiting related-key differentials. In Mehdi Tibouchi and Huaxiong Wang, editors, Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part I, volume 13090 of Lecture Notes in Computer Science, pages 241-271. Springer, 2021.

[20] Alex Biryukov and Ivica Nikolic. Automatic search for related-key differential characteristics in byte-oriented block ciphers: Application to aes, camellia, khazad and others. In Henri Gilbert, editor, Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings, volume 6110 of Lecture Notes in Computer Science, pages 322-344. Springer, 2010.

[21] Pierre-Alain Fouque, Jérémy Jean, and Thomas Peyrin. Structural evaluation of AES and chosen-key distinguisher of 9-round AES-128. In Ran Canetti and Juan A. Garay, editors, Advances in Cryptology - CRYPTO 2013 – 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I, volume 8042 of Lecture Notes in Computer Science, pages 183-203. Springer, 2013.

[22] Carlos Cid, Tao Huang, Thomas Peyrin, Yu Sasaki, and Ling Song. A security analysis of deoxys and its internal tweakable block ciphers. IACR Trans. Symmetric Cryptol., 2017(3):73-107, 2017.

[23] Xiaoyang Dong, Jian Guo, Shun Li, and Phuong Pham. Triangulating rebound attack on aes-like hashing. In Yevgeniy Dodis and Thomas Shrimpton, editors, Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part I, volume 13507 of Lecture Notes in Computer Science, pages 94-124. Springer, 2022.

[24] Ivan Damgård. A design principle for hash functions. In Gilles Brassard, editor, Advances in Cryptology - CRYPTO 1989, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings, volume 435 of Lecture Notes in Computer Science, pages 416-427. Springer, 1989.

[25] Ralph C. Merkle. One way hash functions and DES. In Gilles Brassard, editor, Advances in Cryptology - CRYPTO 1989, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings, volume 435 of Lecture Notes in Computer Science, pages 428-446. Springer, 1989.

[26] Antoon Bosselaers and Bart Preneel, editors. Integrity Primitives for Secure Information Systems, Final Report of RACE Integrity Primitives Evaluation RIPE-RACE 1040, volume 1007 of Lecture Notes in Computer Science. Springer, 1995.

[27] Xiaoyun Wang and Hongbo Yu. How to break MD5 and other hash functions. In Ronald Cramer, editor, Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings, volume 3494 of Lecture Notes in Computer Science, pages 19-35. Springer, 2005.

[28] Xiaoyun Wang, Hongbo Yu, and Yiqun Lisa Yin. Eﬀicient collision search attacks on SHA-0. In Victor Shoup, editor, Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings, volume 3621 of Lecture Notes in Computer Science, pages 1-16. Springer, 2005.

[29] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Finding collisions in the full SHA-1. In Victor Shoup, editor, Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings, volume 3621 of Lecture Notes in Computer Science, pages 17-36. Springer, 2005.

[30] Xiaoyun Wang, Xuejia Lai, Dengguo Feng, Hui Chen, and Xiuyuan Yu. Cryptanalysis of the hash functions MD4 and RIPEMD. In Ronald Cramer, editor, Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings, volume 3494 of Lecture Notes in Computer Science, pages 1-18. Springer, 2005.

[31] Morris J Dworkin. Sha-3 standard: Permutation-based hash and extendable-output functions. 2015.

[32] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Keccak sponge function family main document. Submission to NIST (Round 2), 3(30): 320-337, 2009.

[33] Itai Dinur, Paweł Morawiecki, Josef Pieprzyk, Marian Srebrny, and Michał Straus. Cube attacks and cube-attack- like cryptanalysis on the round-reduced keccak sponge function. In Advances in Cryptology–EUROCRYPT 2015: 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I 34, pages 733-761. Springer, 2015.

[34] Senyang Huang, Xiaoyun Wang, Guangwu Xu, Meiqin Wang, and Jingyuan Zhao. Conditional cube attack on reduced-round keccak sponge function. In Advances in Cryptology–EUROCRYPT 2017: 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30–May 4, 2017, Proceedings, Part II 36, pages 259-288. Springer, 2017.

[35] María Naya-Plasencia, Andrea Röck, and Willi Meier. Practical analysis of reduced-round keccak. In International Conference on Cryptology in India, pages 236-254. Springer, 2011.

[36] Itai Dinur, Orr Dunkelman, and Adi Shamir. New attacks on keccak-224 and keccak-256. In Fast Software Encryption: 19th International Workshop, FSE 2012, Washington, DC, USA, March 19-21, 2012. Revised Selected Papers, pages 442-461. Springer, 2012.

[37] Itai Dinur, Orr Dunkelman, and Adi Shamir. Collision attacks on up to 5 rounds of sha-3 using generalized internal differentials. In International Workshop on Fast Software Encryption, pages 219-240. Springer, 2013.

[38] Ling Song, Guohong Liao, and Jian Guo. Non-full sbox linearization: applications to collision attacks on round reduced keccak. In Advances in Cryptology–CRYPTO 2017: 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part II 37, pages 428-451. Springer, 2017.

[39] Kexin Qiao, Ling Song, Meicheng Liu, and Jian Guo. New collision attacks on round-reduced keccak. In Advances in Cryptology–EUROCRYPT 2017: 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30-May 4, 2017, Proceedings, Part III 36, pages 216-243. Springer, 2017.

[40] Jian Guo, Guohong Liao, Guozhen Liu, Meicheng Liu, Kexin Qiao, and Ling Song. Practical collision attacks against round-reduced sha-3. Journal of Cryptology, 33: 228-270, 2020.

[41] Senyang Huang, Orna Agmon Ben-Yehuda, Orr Dunkelman, and Alexander Maximov. Finding collisions against 4-round sha-3-384 in practical time. IACR Transactions on Symmetric Cryptology, pages 239-270, 2022.

[42] Zhongyi Zhang, Chengan Hou, and Meicheng Liu. Collision attacks on round-reduced sha-3 using conditional internal differentials. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 220-251. Springer, 2023.

[43] Whitfield Diﬀie and Martin E Hellman. Special feature exhaustive cryptanalysis of the nbs data encryption standard. Computer, 10(6): 74-84, 1977.

[44] Kazumaro Aoki and Yu Sasaki. Preimage attacks on one-block md4, 63-step md5 and more. In Selected Areas in Cryptography: 15th International Workshop, SAC 2008, Sackville, New Brunswick, Canada, August 14-15, Revised Selected Papers 15, pages 103-119. Springer, 2009.

[45] Simon Knellwolf and Dmitry Khovratovich. New preimage attacks against reduced sha-1. In Annual Cryptology Conference, pages 367–383. Springer, 2012.

[46] Thomas Espitau, Pierre-Alain Fouque, and Pierre Karpman. Higher-order differential meet-in-the-middle preimage attacks on sha-1 and blake. In Annual Cryptology Conference, pages 683-701. Springer, 2015.

[47] Yu Sasaki and Kazumaro Aoki. Finding preimages in full md5 faster than exhaustive search. In Advances in Cryptology-EUROCRYPT 2009: 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings 28, pages 134-152. Springer, 2009.

[48] Dmitry Khovratovich, Christian Rechberger, and Alexandra Savelieva. Bicliques for preimages: attacks on skein-512 and the sha-2 family. In International Workshop on Fast Software Encryption, pages 244-263. Springer, 2012.

[49] Kazumaro Aoki, Jian Guo, Krystian Matusiewicz, Yu Sasaki, and Lei Wang. Preimages for step-reduced sha-2. In Advances in Cryptology–ASIACRYPT 2009: 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings 15, pages 578-597. Springer, 2009.

[50] Jian Guo, San Ling, Christian Rechberger, and Huaxiong Wang. Advanced meet-in-the-middle preimage attacks: First results on full tiger, and improved results on md4 and sha-2. In Advances in Cryptology-ASIACRYPT 2010: 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings 16, pages 56-75. Springer, 2010.

[51] Zhenzhen Bao, Lin Ding, Jian Guo, Haoyang Wang, and Wenying Zhang. Improved meet-in-the-middle preimage attacks against aes hashing modes. IACR Transactions on Symmetric Cryptology, pages 318-347, 2019.

[52] Paweł Morawiecki and Marian Srebrny. A sat-based preimage analysis of reduced keccak hash functions. Information Processing Letters, 113(10-11): 392-397, 2013.

[53] Pawel Morawiecki, Josef Pieprzyk, Marian Srebrny, and Michal Straus. Preimage attacks on the round-reduced keccak with the aid of differential cryptanalysis. Cryptology ePrint Archive, 2013.

[54] Jian Guo, Meicheng Liu, and Ling Song. Linear structures: applications to cryptanalysis of round-reduced keccak. In Advances in Cryptology–ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I 22, pages 249-274. Springer, 2016.

[55] Rajendra Kumar, Nikhil Mittal, and Shashank Singh. Cryptanalysis of 2 round keccak-384. In International Conference on Cryptology in India, pages 120-133. Springer, 2018.

[56] Ting Li and Yao Sun. Preimage attacks on round-reduced keccak-224/256 via an allocating approach. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 556-584. Springer, 2019.

[57] Xiaoen Lin, Le He, and Hongbo Yu. Improved preimage attacks on 3-round keccak-224/256. IACR Transactions on Symmetric Cryptology, pages 84-101, 2021.

[58] Junling Pei and Lusheng Chen. Preimage attacks on reduced-round keccak hash functions by solving algebraic systems. IET Information Security, 17(3): 394-406, 2023.

[59] Le He, Xiaoen Lin, and Hongbo Yu. Improved preimage attacks on 4-round keccak-224/256. IACR Transactions on Symmetric Cryptology, pages 217-238, 2021.

[60] Itai Dinur. Cryptanalytic applications of the polynomial method for solving multivariate equation systems over gf (2). In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 374-403. Springer, 2021.

[61] Congming Wei, Chenhao Wu, Ximing Fu, Xiaoyang Dong, Kai He, Jue Hong, and Xiaoyun Wang. Preimage attacks on 4-round keccak by solving multivariate quadratic systems. In International Conference on Information Security and Cryptology, pages 195-216. Springer, 2021.

[62] Mahesh Sreekumar Rajasree. Cryptanalysis of round-reduced keccak using non-linear structures. In Progress in Cryptology - INDOCRYPT 2019: 20th International Conference on Cryptology in India, Hyderabad, India, December 15-18, 2019, Proceedings 20, pages 175-192. Springer, 2019.

[63] Fukang Liu, Takanori Isobe, Willi Meier, and Zhonghao Yang. Algebraic attacks on round-reduced keccak. In Information Security and Privacy: 26th Australasian Conference, ACISP 2021, Virtual Event, December 1-3, 2021, Proceedings 26, pages 91-110. Springer, 2021.

[64] Lingyue Qin, Jialiang Hua, Xiaoyang Dong, Hailun Yan, and Xiaoyun Wang. Meet-in-the-middle preimage attacks on sponge-based hashing. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 158-188. Springer, 2023.

[65] Whitfield Diﬀie and Martin E Hellman. New directions in cryptography. In Democratizing Cryptography: The Work of Whitfield Diﬀie and Martin Hellman, pages 365-390. 2022.