



基于 MILP 的 JH-512 杂凑函数中间相遇原像攻击: 突破 13 轮

李铮^{1,2}, 贺一民¹, 闫海伦^{3,4*}, 赵亮², 张建标¹

1. 北京工业大学计算机学院, 北京 100124

2. 四川大学数据安全防护与智能治理教育部重点实验室, 成都 610065

3. 中国科学院大学密码学院, 北京 100049

4. 密码科学技术全国重点实验室, 北京 100878

* 通信作者. E-mail: hailun.yan@ucas.ac.cn

收稿日期: 2025-09-08; 修回日期: 2025-11-05; 接受日期: 2025-12-01; 网络出版日期: 2026-03-20

国家重点研发计划 (批准号: 2024YFB4504800)、国家自然科学基金 (批准号: 62202017, 62202444, 62532013, 61302161, 62271335)、数据安全防护与智能治理教育部重点实验室 (四川大学)、四川大学中央高校基本科研业务费专项资金 (批准号: SCU2023D008)、教育部高等学校博士学科点专项科研基金项目 (批准号: 20130181120076) 和国家密码科学基金 (批准号: 2025NCSF01012) 资助项目

摘要 近年来, 中间相遇 (meet-in-the-middle, MitM) 攻击在高级加密标准 (advanced encryption standard, AES) 类结构和海绵型结构的密码分析中均取得了显著进展. 作为 SHA-3 竞赛的决赛入围者, JH 杂凑函数采用了基于置换的压缩函数结构, 其设计思想与 Even-Mansour 结构相近, 并结合了广义的 AES 设计方法论. 本文针对 JH-512 的置换提出了一种 MitM 攻击建模框架, 并利用混合整数线性规划 (mixed-integer linear programming, MILP) 方法自动搜索, 成功找到了一条覆盖 13 轮的 MitM 特征. 在此基础上, 我们对缩减至 13/42 轮的 JH-512 进行了 MitM 原像攻击, 时间复杂度为 2^{320} , 相比当前已知最佳的 10/42 轮 (时间复杂度 2^{325}) 原像攻击进一步改进. 虽然完整版 JH 依然被认为是安全的, 但本文提出的方法为 JH 及同类原语的密码分析提供了新的思路与工具支持.

关键词 杂凑函数, Even-Mansour 结构, 中间相遇攻击, 原像攻击, JH

1 引言

中间相遇 (meet-in-the-middle, MitM) 攻击作为在时间与空间之间实现策略性权衡的经典方法, 自提出以来一直是对称密码分析的重要工具. 其思想最早可以追溯至 Diffie 与 Hellman 于 1977 年的工作^[1], 随后在二重 DES 与双密钥三重 DES 的语境中被系统化与推广^[2], 从而确立了 MitM 攻击范式在分组密码分析中的基础地位. 在更广泛的多重加密与时间-存储折中研究脉络中, 文献^[3]进一步推动了此类方法论的传播与应用. 近期, 随着海绵 (sponge) 结构与高级加密标准 (advanced encryption standard, AES) 类结构等现代设计的广泛应用, MitM 攻击在原像 (preimage)、碰撞 (collision) 与区分 (distinguishing) 等多种攻击场景下均展现出较强的适应性与扩展性.

引用格式: 李铮, 贺一民, 闫海伦, 等. 基于 MILP 的 JH-512 杂凑函数中间相遇原像攻击: 突破 13 轮. 中国科学: 信息科学, 2026, 56: 868–883, doi: 10.1360/SSI-2025-0389

Li Z, He Y M, Yan H L, et al. MILP-based meet-in-the-middle preimage attack on the JH-512 hash function: breaking 13 rounds. Sci Sin Inform, 2026, 56: 868–883, doi: 10.1360/SSI-2025-0389

围绕如何在复杂结构中降低搜索难度,提高匹配效率与攻击的可行性,研究人员提出并不断完善了若干关键技术分支,形成了相互补充的工具箱. 其一是空间压缩与结构利用: Dunkelman 等在 INDOCRYPT 2007 引入的内部状态猜测技术 (internal state guessing technique)^[4] 通过局部猜测减少了状态的不确定性; Aoki 等在 SAC 2008 提出的拼切技术 (splice-and-cut technique)^[5] 在 MitM 攻击框架中设定中部切口,以少量自由度拼接前后路径,将严格全等改为可局部匹配,从而降低了匹配成本并延伸了可攻击轮数; Sasaki 等在 EUROCRYPT 2009 公开的初始结构分析技术 (initial structure analysis technique)^[6] 则利用了加密初始阶段的已知结构特性进行定制化建模. 其二是匹配能力的增强: Bogdanov 等在 ASIACRYPT 2011 引入并由 Khovratovich 等在 EUROCRYPT 2012 改进的双团技术 (biclique technique)^[7,8] 通过构造前后向计算之间的二维桥接结构,批量抵消中间差异并跨越若干轮,以接近穷举的代价换取微弱的时间优势; Bogdanov 等在 SAC 2010 提出的三子集中间相遇 (3-Subset MitM) 攻击^[9] 将二子集碰撞扩展到三子集以提升灵活性; Aoki 等在 SAC 2008 与 Sasaki 等在 EUROCRYPT 2009 中使用的间接部分匹配 (indirect-partial matching)^[5,6] 与 Canteaut 等在 CRYPTO 2013 提出的中间筛选技术 (sieve-in-the-middle technique)^[10] 通过放宽“完全匹配”的强约束并在匹配前进行过滤,显著降低了候选规模; Fuhr 等在 FSE 2014 提出的匹配盒技术 (match-box technique)^[11] 提供了可扩展的数据组织与对齐手段. 其三是搜索组织与分治: Dinur 等在 CRYPTO 2012 提出的剖分 (dissection) 框架^[12] 以分治视角重塑了搜索流程,在大规模参数下兼顾时间与空间的平衡. 上述技术在 AES 与 AES 类原语上得到了系统验证: Demirci 等在 FSE 2008 提出的 Demirci-Selçuk MitM 对 8 轮 AES 的攻击^[13] 及其后续单密钥改进^[14,15], 配合对 Rijndael 结构的早期研究^[16], 展示了 MitM 攻击与差分 (differential) 和线性 (linear) 等传统分析手段的协同; 相关思路亦被扩展至轻量级分组密码 (如 KTANTAN 与 Piccolo)^[9,17,18], 并由 Isobe 等在 SAC 2012 发展出全子密钥恢复框架 (all-subkeys recovery frameworks)^[19]. 总体来看,上述分支分别在可拼接性、可匹配性、可分治性等方面各有侧重,能力互补,但也不可避免地面临模型规模、预处理代价与内存占用等方面的权衡问题.

MitM 攻击与差分分析的结合经历了从启发式到形式化建模的演进. Knellwolf 等在 CRYPTO 2012 和 Espitau 等在 CRYPTO 2015 的工作^[20,21] 首次系统提出差分 MitM 攻击思路,并在框架层面给出可复用的建模策略; Boura 等在 CRYPTO 2023 则对现代差分 MitM 攻击进行了更为严谨的形式化^[22]. 近年来,该方向围绕差分可达性刻画、截断与模式化传播、匹配概率估计与优化等关键环节持续推进: 包括对既有方法的重新审视^[23]、Ahmadian 等在 EUROCRYPT 2024 的改进技术^[24], 以及关于轻量级设计^[25,26] 与可调整结构 (tweakable constructions)^[27,28] 的新案例研究. 这些工作共同提升了差分 MitM 攻击的可解释性与可迁移性,但在模型紧致度与求解可行性的折中等方面仍需在具体原语上进行结构化权衡与实现.

为了减少对手工推导与试错过程的依赖,面向对称密码分析的自动化研究得到了快速发展. Bouillaguet 等^[29] 的早期尝试聚焦于 AES 及同类分组密码,探索将匹配与搜索过程程序化; 随后, Demirci-Selçuk 型中间相遇攻击模型在后续研究中得到了进一步分析与改进^[30]. 在此基础上, Shi 等^[31] 首次以带约束的程序化方式对 Demirci-Selçuk 中间相遇攻击进行编码,并进一步在 GIFT 等密码原语上以混合整数线性规划 (mixed integer linear programming, MILP) 实现 3-Subset MitM 攻击^[32]. Bao 等在 EUROCRYPT 2021 中将 MILP 方法推广至 AES 类结构的杂凑函数,实现了全自动的原像中间相遇攻击^[33]. 在该模型基础上, Dong 等与 Bao 等分别在 CRYPTO 2021 和 CRYPTO 2022 发表了关于密钥恢复攻击与碰撞攻击扩展的研究成果^[34,35]. 此外,文献 [36,37] 进一步给出了模型的简化方法,并探讨了其在经典与量子计算场景下的应用. 近期工作将自动化 MitM 攻击推广到更多结构,包括 Shi 等在 EUROCRYPT 2023 上面向 SKINNY 分组密码的完整全自动 Demirci-selçuk MitM 密码分析^[38]、Hou 等在 ASIACRYPT 2023 上面向一般 Feistel 网络的分析^[39]、面向具有二进制矩阵线性层的原语^[40], 以及进一步的自动化与差分 MitM 攻击工具^[41,42]. 在更广的应用侧, MitM 攻击亦与其他攻击家族形成互补: 例如 He 等在 EUROCRYPT 2024 的序列密码立方攻击 (cube attack) 中实现了大规模超级多项式 (superpoly) 恢复^[43], Dong 等在 CRYPTO 2024 提出了基于海绵结构的通用框架^[44]. 与此同时,深度学习辅助密码分析在特征筛选、参数搜索与模式识别方面展现出互补潜力,形成与自动化 MitM 攻击相呼应的工具链^[45~48]. 这些结果表明,自动化 MitM 攻击的优势在于其高可移植性与系统化的探索能力,但其有效性高度依赖于对原语结构的精细建模与对求解器的友好度,并需要在二者之间取得平衡.

表 1 JH-512 及其他典型 AES 类结构杂凑函数原像攻击结果. 表中加粗条目表示本文所达到的最优攻击结果, 包括可攻击的最大轮数及其对应的时间复杂度.

Table 1 Preimage attack results on JH-512 and other representative AES-like hash functions. Entries in bold denote the best results achieved in this work, including the maximum number of attacked rounds and the corresponding time complexities.

Target	Rounds	Time complexity	References
JH-512	10/42	2^{325}	[56]
	11/42	2^{320}	Section 4.2
	12/42	2^{320}	Section 4.2
	13/42	2^{320}	Section 4.1
AES-128	7/10	2^{120}	[59]
	7/10	2^{112}	[52]
Rijndael-192	8/12	2^{176}	[52]
Rijndael-256	9/14	2^{248}	[33]

在 AES 类杂凑函数与 Sponge 结构中, MitM 攻击被用于构造 Nostradamus 攻击与更深入的原像攻击^[49~51], 并扩展至单向函数设定^[52]. 此外, Dong 等在 CRYPTO 2024 提出的面向 Sponge 结构的通用 MitM 攻击框架^[44] 为跨原语对齐与统一评估提供了方法基础, 在实践中可与差分 MitM 攻击的形式化建模^[22~25, 27] 形成配套关系, 进一步提升自动化搜索的覆盖度与可靠性.

针对 JH 杂凑函数^[53] 的研究表明, 其置换与压缩结构兼具较强的代数与线性层耦合特性, 早期工作总结了其模式与结构性质^[54], 并给出了针对 42 轮内部置换的区分器 (时间复杂度为 2^{304}), 同时, Naya-Plasencia 等在 ASIACRYPT 2011 的工作指出该思路可扩展至完整压缩函数^[55]. 在原像方向, 已有结果在 10/42 轮上达到时间复杂度为 2^{325} 的原像攻击^[56]. 此外, 已有工作对 JH 及第三代安全杂凑算法 (secure hash algorithm 3, SHA-3) 部分候选杂凑函数的安全性分析与攻击结果进行了系统研究^[57]. 然而, 据我们所知, 超过 10 轮的 JH-512 原像攻击尚未公开. 综合自动化 MitM 攻击与差分 MitM 攻击的近期建模思想^[23, 24, 34, 50, 51], 一个自然的问题是: 能否在保持建模严谨与求解可行的前提下, 构建对求解器友好的 MILP 自动化 MitM 攻击框架, 系统挖掘更长轮数的 JH 原像攻击并降低其复杂度?

为了解决上述问题, 本文设计了面向 JH-512 置换与压缩函数的可扩展 MILP 自动化 MitM 攻击模型, 实现了对 JH-512 算法传播与匹配关系的统一刻画, 并利用 Gurobi 求解器^[58] 进行 MitM 特征搜索与复杂度评估. 表 1 汇总了当前针对 JH 及其他典型 AES 类结构杂凑函数原像攻击的结果, 其中包括我们的新进展.

如表 1 所示, 首先在 11/42 与 12/42 轮上给出自动化 MitM 原像攻击, 时间复杂度均为 2^{320} . 在此基础上, 进一步将攻击轮数提升至 13/42 轮, 并在相同复杂度下实现原像攻击. 与既有的 10/42 轮 (时间复杂度 2^{325})^[56] 的结果相比, 我们的三项结果均在更长轮数上取得更优的时间复杂度. 详细攻击流程与参数见第 4 节.

本文贡献. 本文工作表明, 针对 JH-512 算法, 通过 MILP 自动化搜索 MitM 特征并得出对应的原像攻击是可行的. 本文的主要贡献包括如下.

- **面向 JH 的 MILP 自动化 MitM 攻击模型.** 我们提出可扩展的建模框架, 针对半字节 (nibble) 引入 2 比特颜色编码 (b, r); 采用基于凸包不等式的线性层建模与基于索引的置换约束, 以分层方式描述跨 $S/L/P$ 层的传播; 将颜色的可用性与选择解耦, 以正确计入自由度; 同时结合基于 CP-kernel 的匹配约束与最大化 $\min\{DoB_T, DoR_T, DoM_T\}$ 的目标函数.

- **11 和 12 轮的自动化 MitM 攻击与参数化评估.** 基于上述模型并借助 Gurobi^[58], 系统化得到 11/42 与 12/42 轮的原像攻击: 均满足 $DoB_T = DoR_T = 48, DoM_T = 64$, 时间复杂度为 2^{320} .

- **一个 13 轮 MitM 特征与首个 13/42 轮 JH-512 原像攻击.** 在 11 和 12 轮的建模与参数化基础上, 我们进一步找到一个 13 轮特征, 其 $DoB_T = DoR_T = 48$ 且 $DoM_T = 64$, 据此给出对 JH-512 的 13 轮中间相遇原像攻击, 时间复杂度为 2^{320} . 该结果将先前公开的最佳结果 (10/42 轮, 时间复杂度 2^{325})^[56] 推进至 13/42 轮并实

现更低的复杂度,进一步缩小与完整 42 轮设计之间的差距.

本文后续组织如下:第 2 节给出符号约定与预备知识;第 3 节详细介绍基于 MILP 的自动化 MitM 攻击建模方法;第 4 节给出针对 11, 12 和 13 轮 JH-512 算法的原像攻击设计、参数与复杂度评估;第 5 节对工作进行总结和展望.

2 预备知识

本节给出本文所用的符号约定与基本记法,并简要回顾 JH 的压缩函数与内部置换结构,同时提供对 MitM 攻击的基本介绍,以便为后续建模与攻击描述奠定统一的语义基础.默认 $\parallel$ 表示串接, $\oplus$ 表示逐比特异或;半字节 (nibble) 指 4 比特单元.

2.1 符号说明

为避免歧义,对涉及状态、消息、分支与自由度等对象的符号做如下统一说明.

字:一段比特串.

A^i : 字 A 的第 i 比特.若 A 为 m 比特,则 $A = A^0 \parallel A^1 \parallel A^2 \parallel \dots \parallel A^{m-1}$.

d : 维度参数.一个 d 维分组包含 2^d 个 4 比特单元.

$C_d^{(r)}$: E_d 的第 r 轮常量, $0 \leq r \leq 6 \times (d-1)$.每个 $C_d^{(r)}$ 均为一个 2^d 比特的常数.

h : 杂凑值的比特数.在 JH-512 中, $h = 1024$.

m : 消息分组 $M^{(i)}$ 的比特数.在 JH-512 中, $m = 512$.

N : 填充后消息的分组数.

$M^{(i)}$: 第 i 个消息分组,长度为 m 比特.

$H^{(i)}$: 第 i 个杂凑值,长度为 h 比特.

$H_L^{(i)}, H_R^{(i)}$: $H^{(i)}$ 的左/右半部,各由 512 比特组成.

$A^{(i)}$: 第 i 轮 E_d 的输入状态.

$A_L^{(i)}, A_R^{(i)}$: $A^{(i)}$ 的左/右半部,各由 512 比特组成.

$B^{(i)}$: 第 i 轮 E_d 的输出状态.

$B_L^{(i)}, B_R^{(i)}$: $B^{(i)}$ 的左/右半部,各由 512 比特组成.

n : 匹配域中的单元即半字节的数量.

DoB_T : 匹配点处前向分支的剩余自由度,单位为半字节.

DoR_T : 匹配点处后向分支的剩余自由度,单位为半字节.

DoM_T : 匹配点处可用的总匹配自由度,单位为半字节.

2.2 JH 压缩函数

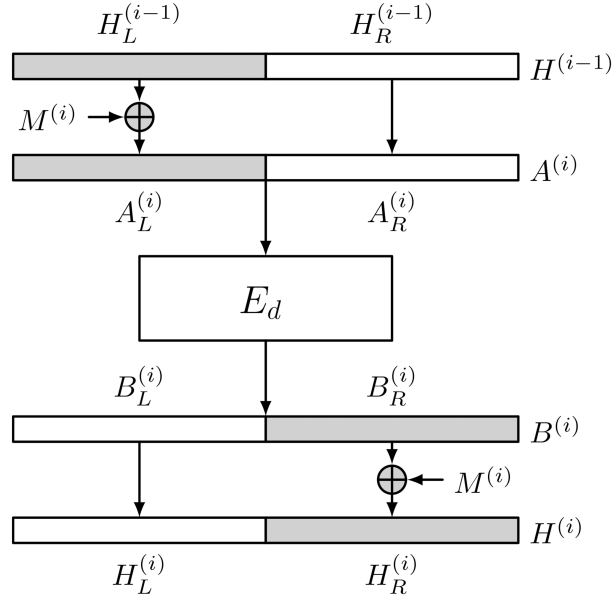
本小节简要回顾本文攻击所需的 JH 压缩函数 F_d 与内部置换 E_d 的局部细节,遵循规范中的定义 (完整规范参见文献 [53]).

2.2.1 压缩函数 F_8

设 F_d 是由 2^{d+2} 比特置换 E_d 导出的压缩函数.给定 2^{d+2} 比特杂凑值 $H^{(i-1)}$ 与 2^{d+1} 比特消息分组 $M^{(i)}$, 压缩为

$$H^{(i)} = F_d(H^{(i-1)}, M^{(i)}).$$

压缩函数 F_d 的结构如图 1 所示.

图 1 压缩函数 F_d .Figure 1 The compression function F_d .

在 JH-512 的场景中, $d = 8$, 因此 F_8 将 1024 比特状态与 512 比特消息分组映射为新的 1024 比特状态. 更具体地, 记 $A^{(i)}, B^{(i)} \in \{0, 1\}^{1024}$, 则

$$\begin{aligned}
 \text{(a)} \quad A^{(i),j} &= \begin{cases} H^{(i-1),j} \oplus M^{(i),j}, & 0 \leq j \leq 511; \\ H^{(i-1),j}, & 512 \leq j \leq 1023; \end{cases} \\
 \text{(b)} \quad B^{(i),j} &= E_8(A); \\
 \text{(c)} \quad H^{(i),j} &= \begin{cases} B^j, & 0 \leq j \leq 511; \\ B^j \oplus M^{(i),j-512}, & 512 \leq j \leq 1023. \end{cases}
 \end{aligned}$$

在半状态层面, 可等价写为

$$\begin{cases} A_L^{(i)} = H_L^{(i-1)} \oplus M^{(i)}, \\ H_R^{(i)} = B_R^{(i)} \oplus M^{(i)}. \end{cases} \quad (1)$$

从而消去 $M^{(i)}$, 得到后文匹配建模中使用的关键关系

$$B_R^{(i)} = H_R^{(i)} \oplus A_L^{(i)} \oplus H_L^{(i-1)}. \quad (2)$$

2.2.2 置换函数 E_8

置换 E_d 是作用于 2^d 个 4 比特半字节 (总计 2^{d+2} 比特) 上的 $6(d-1)$ 轮代换 - 置换网络 (substitution-permutation network, SPN). 每轮由三部分依次构成.

- S 盒 (substitution box, S-Box) 层: 对第 i 个半字节 a_i 应用 S 盒 S_0 或 S_1 , 选择由轮常量 $C_d^{(r)}$ 的第 i 位决定.
- 线性层: 对相邻半字节对 (v_{2i}, v_{2i+1}) 施加 2×2 的最大距离可分 (maximum distance separable, MDS) 线性变换 L , 得到 (w_{2i}, w_{2i+1}) .
- 置换层: 对半字节向量施加固定置换 P_d , 以在不同维度间轮换扩散方向.

设输入/输出半字节向量分别为 $A = (a_0, \dots, a_{2^d-1})$ 和 $B = (b_0, \dots, b_{2^d-1})$. 第 r 轮 (其轮常量记为 $C_r^{(d)} = (C_r^{(d),0} \parallel \dots \parallel C_r^{(d),2^d-1})$) 的函数为

$$\begin{aligned} \text{(a)} \quad v_i &= \begin{cases} S_0(a_i), & C_r^{(d),i} = 0, \\ S_1(a_i), & C_r^{(d),i} = 1, \end{cases} \quad 0 \leq i \leq 2^d - 1; \\ \text{(b)} \quad (w_{2i}, w_{2i+1}) &= L(v_{2i}, v_{2i+1}), \quad 0 \leq i \leq 2^{d-1} - 1; \\ \text{(c)} \quad (b_0, b_1, \dots, b_{2^d-1}) &= P_d(w_0, w_1, \dots, w_{2^d-1}). \end{aligned}$$

轮常量按照 JH 规范所述, 通过迭代一个简化实例生成:

$$\begin{aligned} C_0^{(d)} &= \lfloor (\sqrt{2} - 1) \cdot 2^{2^d} \rfloor, \\ C_r^{(d)} &= R_{d-2}(C_{r-1}^{(d)}, 0), \quad 1 \leq r < 6(d-1). \end{aligned}$$

针对 $d=8$ 的显式轮常量列表与 1024 比特串与半字节索引的比特级分组/解组细节, 请参见 JH 规范^[53].

置换 P_8 . 固定置换 P_d 对 2^d 个半字节进行重排, 使得跨轮时线性层沿不同维度起作用. 本文采用 JH 规范中的标准映射^[53].

线性变换 L . L 定义在由不可约多项式 $x^4 + x + 1$ 确定的有限域 $\text{GF}(2^4)$ 上. 对于 4 比特字 A, B :

$$(C, D) = (5 \cdot A + 2 \cdot B, 2 \cdot A + B),$$

其中 “ $\cdot$ ” 表示域乘法. 为简洁起见, 省略比特级展开.

S 盒. 使用两个 4 比特 S 盒 S_0, S_1 , 逐半字节的选择由轮常量比特控制.

2.3 JH-512 杂凑函数

本小节回顾 JH-512 杂凑函数的外层流程: 填充、解析、初始化、压缩与截断输出.

填充和解析. 给定长度为 l 的消息 M , 在其后追加单个 “1” 比特, 再追加若干 “0” 比特, 并追加 l 的 128 位大端编码, 使结果总长度为 512 的整数倍. 将填充后的消息拆分为 N 个 512 比特分组 $M^{(1)}, \dots, M^{(N)}$.

初始化. 令 1024 比特向量 $H^{(-1)}$ 在其前两个字节中编码期望的输出大小 (对于 JH-512: 0x0200), 其余位置为零. 设 $M^{(0)} = 0$. 初始状态为

$$H_0^{(-1),0} \parallel H_0^{(-1),1} \parallel \dots \parallel H_0^{(-1),15} = 0\text{x}0200.$$

压缩. 对于 $i = 1, \dots, N$, 应用 JH 压缩函数 F_8 :

$$H^{(i)} = F_8(H^{(i-1)}, M^{(i)}).$$

截断输出. 将最终杂凑值 $H^{(N)}$ 的后 512 比特截断, 输出为 512 比特的消息摘要:

$$\text{JH-512}(M) = H^{(N),512} \parallel H^{(N),513} \parallel \dots \parallel H^{(N),1023}.$$

2.4 MitM 攻击流程

MitM 攻击的核心思想是: 在压缩函数或分组算法的迭代结构选定一个中间匹配点 r_m , 将计算路径一分为二: 从已知起点 r_s 向中间匹配点 r_m 正向推进的称为前向块 (forward chunk), 自靠近末端处向 r_m 反向推进的称为后向块 (backward chunk). 在 r_m 处对两侧产生的中间信息施加一致性约束以完成匹配, 从而将整体搜索转化为两个规模较小的枚举并在中间相遇.

为使两段计算彼此独立、可并行扩展, 分别在前/后向块中选择互不相交的中性字 (neutral words), 即各自可自由枚举且不依赖对方未知量的一组消息/状态单元.

典型流程如下.

- (1) 选定匹配点 r_m , 将路径划分为前向与后向两段.
- (2) 前向: 对前向块的中性字进行枚举, 推进到 r_m 并记录相应的中间信息 (形成表 B).
- (3) 后向: 对后向块的中性字进行枚举, 回溯到 r_m 并记录相应的中间信息 (形成表 R).
- (4) 匹配: 在 r_m 处根据预先设定的中间一致性条件, 在 B 与 R 中查找匹配对.
- (5) 验证: 由匹配对还原完整输入并计算输出, 与目标值比对以确认有效解.

设计要点是在保证两侧独立性的前提下, 提高前/后向的可枚举自由度, 并在匹配点设置足够的约束以有效筛选候选; 当两侧自由度与匹配规模达到较好平衡时, 可获得更优的整体复杂度.

3 基于 MILP 的自动化 MitM 攻击模型

本节给出 JH 的 MitM 原像攻击的自动化 MILP 建模框架. 遵循常见的颜色编码思路^[33,34], 为每个半字节 (nibble) 赋予 4 种颜色 (■, ■, ■, □) 之一, 并在半字节层面刻画跨 $S/L/P$ 层的属性传播与匹配约束, 最终以最大化 $\min\{DoB_T, DoR_T, DoM_T\}$ 为目标, 以最小化攻击的时间复杂度.

3.1 编码方案

由于在 MILP 模型中需要对 4 种颜色进行建模, 我们采用 2 比特编码方法, 即每个半字节以二元向量 (b, r) 表示颜色.

- 灰色 ■: $(1, 1)$, 全局常量半字节.
- 红色 ■: $(0, 1)$, 由起始状态中的 ■ 半字节与 ■ 半字节确定的半字节.
- 蓝色 ■: $(1, 0)$, 由起始状态中的 ■ 半字节与 ■ 半字节确定的半字节.
- 白色 □: $(0, 0)$, 同时依赖 ■ 与 ■ 的半字节.

记第 r 轮各层输出状态如下:

- 经过 S 盒层后的 $(bS_{i,j}^r, rS_{i,j}^r)$;
- 经过线性变换层 L 后的 $(bL_{i,j}^r, rL_{i,j}^r)$;
- 经过置换层 P_8 后的 $(bP_{i,j}^r, rP_{i,j}^r)$.

其中, $r \in \{1, \dots, Rounds\}$ ($Rounds$ 为总轮数), $i \in \{0, 1\}$ 为行索引, $j \in \{0, 1\}$ 为列索引. 一个 1024 比特状态被视为 256 个 4 比特半字节的二维网格 (2 行 128 列), 按列优先顺序排列. 为表述初始着色, 记 $r = 0$ 表示第 1 轮 S 层之前的初值层.

除非另行说明, 下述颜色指示变量均为 0-1 变量.

3.2 初始状态与边界条件建模

初始状态. 将初始状态的右半部 (列 64, $\dots$, 127) 设为 ■ (全局常量):

$$\begin{cases} bS_{i,j}^0 = 1, & 0 \leq i \leq 1, 64 \leq j \leq 127, \\ rS_{i,j}^0 = 1, & 0 \leq i \leq 1, 64 \leq j \leq 127. \end{cases} \quad (3)$$

同时要求左半部 (列 0, $\dots$, 63) 至少包含一个 ■ 与一个 ■, 以避免退化并为双向提供最小自由度:

$$\begin{cases} 1 \leq \sum_{i=0}^1 \sum_{j=0}^{63} bS_{i,j}^0 \leq 2 \times 64 - 1, \\ 1 \leq \sum_{i=0}^1 \sum_{j=0}^{63} rS_{i,j}^0 \leq 2 \times 64 - 1. \end{cases} \quad (4)$$

输出边界. 在颜色层面, 将压缩函数 F_8 的输出段建模如下.

(1) 输出状态的左半部分 (512 比特) 不受控制 ($\square$), 即两侧均未知且不参与匹配:

$$\begin{cases} bS_{i,j}^{Rounds} = 0, \\ rS_{i,j}^{Rounds} = 0, \end{cases} \quad 0 \leq i \leq 1, 0 \leq j \leq 63. \quad (5)$$

(2) 输出状态的右半部分 (512 比特) 等价于初始状态的左半部分 (由 (2) 式可得到 ($B_R = A_L$) 的颜色等价), 因此,

$$\begin{cases} bS_{i,j}^{Rounds} = bS_{i,j-64}^0, \\ rS_{i,j}^{Rounds} = rS_{i,j-64}^0, \end{cases} \quad 0 \leq i \leq 1, 64 \leq j \leq 127. \quad (6)$$

初始“可用性 – 选择”解耦. 为刻画初始状态中“某颜色在该单元可用”与“实际被选作自由度来源”之间的差异, 引入两个彼此独立的变量 $bI_{i,j}^0$ 与 $rI_{i,j}^0$, 并施加如下约束:

$$\begin{cases} bI_{i,j}^0 \leq bS_{i,j}^0, \\ rI_{i,j}^0 \leq rS_{i,j}^0, \\ bS_{i,j}^0 + rS_{i,j}^0 + bI_{i,j}^0 \leq 2, \\ bS_{i,j}^0 + rS_{i,j}^0 + rI_{i,j}^0 \leq 2, \end{cases} \quad (7)$$

以禁止越域选择与虚增自由度.

3.3 属性传播建模

S 盒层 S . 在颜色层面, S 盒不引入跨侧依赖与自由度消耗, 故作为颜色透传, 无需引入额外约束 (即 (bS^{Rounds}, rS^{Rounds}) 由上一层直接给定), 或视作恒等.

线性层 L . 线性层 L 的传播约束可通过先枚举所有可能的传播模式 (图 2), 再应用凸包 H-表示技术 (convex hull H-representation technique) [60] 以推导出相应的不等式约束.

对于第 r 轮的每一列 j , 记已消耗的自由度为 DoB_j^r, DoR_j^r , 则有

$$\begin{cases} bS_{0,j}^r + bS_{1,j}^r + DoB_j^r \geq bL_{0,j}^r + bL_{1,j}^r, \\ rS_{0,j}^r + rS_{1,j}^r + DoR_j^r \geq rL_{0,j}^r + rL_{1,j}^r, \\ DoB_j^r \geq 0, \\ DoR_j^r \geq 0. \end{cases} \quad (8)$$

直观上, 当 L 层输出端需要的着色超过输入端可提供的数量时, 差额由自由度消耗项承担 (可包含对中性元素施加的线性约束成本).

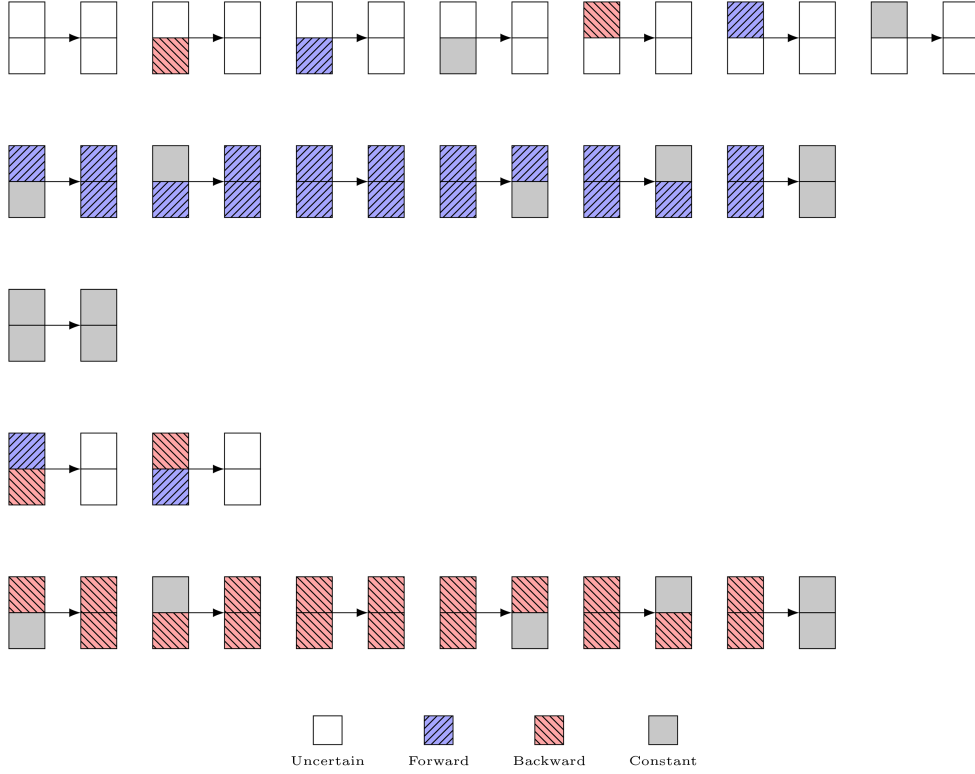
置换层 P . 记 $p[i]$ 为 P_8 的半字节索引映射, 则

$$\begin{cases} bP_{i \bmod 2, \lfloor i/2 \rfloor}^r = bL_{p[i] \bmod 2, \lfloor p[i]/2 \rfloor}^r, \\ rP_{i \bmod 2, \lfloor i/2 \rfloor}^r = rL_{p[i] \bmod 2, \lfloor p[i]/2 \rfloor}^r, \\ bS_{i \bmod 2, \lfloor i/2 \rfloor}^{r+1} = bP_{i \bmod 2, \lfloor i/2 \rfloor}^r, \\ rS_{i \bmod 2, \lfloor i/2 \rfloor}^{r+1} = rP_{i \bmod 2, \lfloor i/2 \rfloor}^r. \end{cases} \quad (9)$$

3.4 匹配阶段建模

以最后一轮 L 层为匹配边界, 分别用二元变量 $BL_{i,j}$ (边界之前) 与 $AL_{i,j}$ (边界之后) 刻画该状态是否为非 $\square$ (即 $\vee$ 运算):

$$BL_{i,j} = (bS_{i,j}^{Rounds} \vee rS_{i,j}^{Rounds}), \quad AL_{i,j} = (bS_{i,j}^{Rounds} \vee rS_{i,j}^{Rounds}).$$

图 2 (网络版彩图) 前向计算中线性变换层 L 的有效着色方案.Figure 2 (Color online) Valid coloring schemes for the linear transformation layer L in the forward computation.

可通过下述标准线性不等式来实现:

$$\begin{cases} BL_{i,j} \geq bS_{i,j}^{Rounds}, \\ BL_{i,j} \geq rS_{i,j}^{Rounds}, \\ BL_{i,j} \leq bS_{i,j}^{Rounds} + rS_{i,j}^{Rounds}, \end{cases} \quad (10)$$

$$\begin{cases} AL_{i,j} \geq bS_{i,j}^{Rounds}, \\ AL_{i,j} \geq rS_{i,j}^{Rounds}, \\ AL_{i,j} \leq bS_{i,j}^{Rounds} + rS_{i,j}^{Rounds}. \end{cases} \quad (11)$$

对第 j 列匹配自由度的汇总为

$$\sum_i BL_{i,j} + \sum_i AL_{i,j} \geq DoM_j. \quad (12)$$

进一步用两个二元指示变量刻画阈值事件:

- $DoM_j^{eq4} = 1$, 当且仅当第 j 列中恰有 4 个半字节匹配;
- $DoM_j^{ge3} = 1$, 当且仅当第 j 列中至少有 3 个半字节匹配.

线性约束为

$$\begin{cases} 4 \times DoM_j^{eq4} - DoM_j \leq 0, \\ DoM_j^{eq4} - DoM_j \geq -3, \end{cases} \quad (13)$$

$$\begin{cases} 3 \times DoM_j^{ge3} - DoM_j \leq 0, \\ 2 \times DoM_j^{ge3} - DoM_j \geq -2. \end{cases} \quad (14)$$

表 2 第 j 列的匹配自由度.
Table 2 Degrees of matching of the j -th column.

DoM_j	DoM_j^{eq4}	DoM_j^{ge3}
0, 1, 2	0	0
3	0	1
4	1	1

从而该列的匹配自由度贡献为 $DoM_j^{eq4} + DoM_j^{ge3}$, 如表 2 所示. 累加得到总匹配自由度

$$DoM_T = \sum_j (DoM_j^{eq4} + DoM_j^{ge3}). \quad (15)$$

3.5 自由度统计与变量域

自由度统计. 初始 (可被选择的) 自由度与传播过程消耗满足

$$\begin{cases} \sum_i \sum_j bI_{i,j}^0 - \sum_r \sum_j DoB_j^r = DoB_T, \\ \sum_i \sum_j rI_{i,j}^0 - \sum_r \sum_j DoR_j^r = DoR_T. \end{cases} \quad (16)$$

变量域. 为避免语义歧义, 给出关键变量的域约束 (求解时可设为二元/整数):

- $bS, bL, bP, rS, rL, rP, bI^0, rI^0, BL, AL, DoM^{eq4}, DoM^{ge3} \in \{0, 1\}$;
- $DoB_{i,j}^r, DoR_{i,j}^r, DoM_T, DoB_T, DoR_T \in \mathbb{Z}_{\geq 0}$;
- $DoM_j \in \{1, 2, 3, 4\}$.

3.6 目标函数

依据第 2 节的单元粒度分析, 并采用本节传播与匹配模型定义的 DoB_T, DoR_T, DoM_T , 推导攻击的计算时间复杂度. 所有计数均以半字节为单位.

(1) 蓝色列表. 通过在既定约束下改变其中性半字节, 蓝色前向块在匹配半字节上枚举 $2^{4 \times DoB_T}$ 个边界取值, 将其存入列表 B .

(2) 红色列表. 同理, 红色前向块枚举 $2^{4 \times DoR_T}$ 个边界取值, 将其存入列表 R .

(3) 匹配过滤器. 在 DoM_T 个半字节上的匹配将 $B \times R$ 的成对组合过滤至期望的 $2^{4 \times (DoB_T + DoR_T - DoM_T)}$ 个剩余对.

(4) 内层循环时间. 构建 B 与 R 的代价为 $2^{4 \times DoB_T} + 2^{4 \times DoR_T}$, 遍历剩余对的代价为 $2^{4 \times (DoB_T + DoR_T - DoM_T)}$. 因此内层循环时间为

$$2^{4 \times \max\{DoB_T, DoR_T, DoB_T + DoR_T - DoM_T\}} = 2^{4 \times (DoB_T + DoR_T - \min\{DoB_T, DoR_T, DoM_T\})}.$$

(5) 从内层循环到完整的 n 单元匹配. n 表示匹配域中单元即半字节的数量. 一次内层循环后, 剩余对的期望数为

$$2^{4 \times (DoB_T + DoR_T - DoM_T)}.$$

这些对在其余 $n - DoM_T$ 个半字节上同时满足完全匹配的概率为

$$2^{-4 \times (n - DoM_T)}.$$

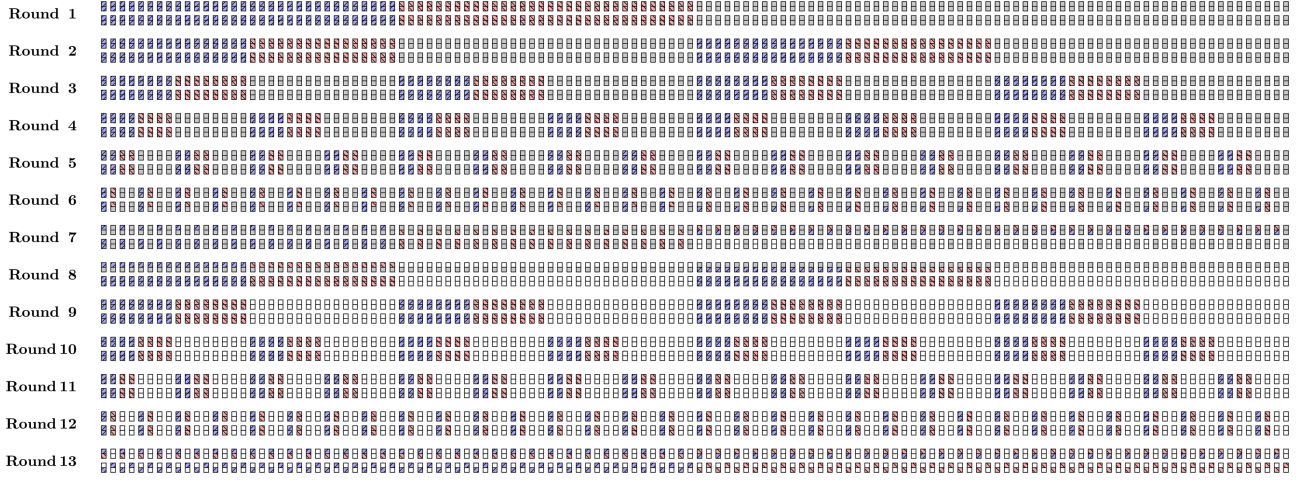


图3 (网络版彩图) 13轮 JH-512 的中间相遇特征.

Figure 3 (Color online) 13-round MitM characteristic of JH-512.

因此每次内层循环得到完整匹配的期望数为

$$2^{4 \times (DoB_T + DoR_T - DoM_T)} \cdot 2^{-4 \times (n - DoM_T)} = 2^{4 \times (DoB_T + DoR_T - n)}.$$

为获得 1 个完整匹配, 所需重复次数为其倒数:

$$2^{4 \times (n - DoB_T - DoR_T)}.$$

则总时间复杂度为

$$2^{4 \times (n - DoB_T - DoR_T)} \cdot 2^{4 \times (DoB_T + DoR_T - \min\{DoB_T, DoR_T, DoM_T\})} = 2^{4 \times (n - \min\{DoB_T, DoR_T, DoM_T\})}.$$

由于攻击的计算复杂度为 $2^{4 \times (n - \min\{DoB_T, DoR_T, DoM_T\})}$, 为尽可能将其最小化, 应当最大化集合 $\{DoB_T, DoR_T, DoM_T\}$ 中的最小值.

$$\begin{cases} Obj_T \geq 1, \\ Obj_T \leq DoB_T, \\ Obj_T \leq DoR_T, \\ Obj_T \leq DoM_T. \end{cases} \quad (17)$$

因此, 在 MILP 中以最大化 Obj_T 为目标, 即可等价最小化攻击的时间复杂度.

4 JH 算法原像攻击

本节基于第 3 节提出的自动化 MitM 攻击模型, 给出 JH-512 算法的缩减轮数原像攻击结果. 首先对 13 轮结果提供完整的攻击流程, 随后给出 11 和 12 轮的 MitM 特征与对应的原像攻击复杂度评估.

4.1 13 轮 JH-512 算法原像攻击

在第 3 节自动化 MitM 攻击模型的基础上, 采用 MILP 进行建模, 并由 Gurobi 优化器 (Gurobi optimizer) [58] 求解, 得到如下的 13 轮 MitM 特征: **13 轮特征**. 如图 3 所示, 其中 $DoB_T = DoR_T = 48, DoM_T = 64$.

该 13 轮 MitM 特征的扩散验证了第 3 节中给出的规则. JH 杂凑函数的初始状态记为 $H^{(-1)}$, 其中前两个字节设置为消息摘要长度并以 0 填充. 将 $H^{(-1)}$ 的左半部分与第一块消息 $M^{(0)}$ 进行异或 (XOR) 后, E_d 的输

入状态即为攻击的起始状态. 该 13 轮 MitM 特征的起始状态包含 64 个  半字节, 64 个  半字节, 128 个  半字节.

对于 13 轮线性变换 L 的输入状态, 其中包含 64 个  半字节, 64 个  半字节, 128 个  半字节. 鉴于消息与 E_d 的输出状态右半部分的异或, 以及置换 P_d , 除了改变半字节的位置外不引入任何变化, 状态 $P_d^{-1}(H_R^{(i)} \oplus A_L^{(0)} \oplus H_L^{(i-1)})$ 亦包含 64 个  半字节, 64 个  半字节, 128 个  半字节. 对于线性变换 L , 左侧 64 列中的每一列均具有 1 个半字节.

为消除歧义, 本文在匹配点 r_m (取为待攻击轮数 $Rounds$ 的最后一轮线性层 L 的输入) 处, 对 B_L 与 R_L 作如下定义, 并与算法 1 中的列表 B 与列表 R 保持一致.

算法 1 13 轮 JH-512 算法原像攻击.

输入: 杂凑值 $\mathcal{T}$;

输出: 一个可能的原像 $\mathcal{P}$;

```

1: for  $2^{\zeta}$  种  $A^{(0)}$  中 128 个自由灰色  半字节的取值 do
2:   /* 预计算阶段 */
3:   for  $2^{4 \times DoB_T} = 2^{192}$  种  $A^{(0)}$  中蓝色  半字节  $v_B$  的取值 do
4:     向前计算至第 13 轮线性变换  $L$  的输入状态;
5:     将该状态存入表  $B$ ;
6:   end for
7:   for  $2^{4 \times DoR_T} = 2^{192}$  种  $A^{(0)}$  中红色  半字节  $v_R$  的取值 do
8:     向前计算至第 13 轮线性变换  $L$  的输入状态;
9:     将该状态存入表  $R$ ;
10:  end for
11: /* 匹配阶段 */
12: flag  $\leftarrow$  0;
13: for 每一对由  $B$  中元素  $B_L^{(12)}$  与  $R$  中元素  $R_L^{(12)}$  组成的二元组  $(B_L^{(12)}, R_L^{(12)})$  do
14:   基于关系  $L(B_L^{(12)}) = L(R_L^{(12)}) = P_d^{-1}(H_R^{(i)} \oplus A_L^{(0)} \oplus H_L^{(i-1)})$  进行匹配测试;
15:   if 匹配成立 then
16:     flag  $\leftarrow$  1;
17:     break
18:   else
19:     continue
20:   end if
21: end for
22: if flag = 1 then
23:   还原完整的  $A^{(0)}$ ;
24:   计算  $t$  比特杂凑值  $\mathcal{T}'$ ;
25:   if  $\mathcal{T}' = \mathcal{T}$  then
26:      $\mathcal{P} \leftarrow A^{(0)}$ ;
27:   else
28:     continue
29:   end if
30: end if
31: end for
32: 返回  $\mathcal{P}$ .
```

记 $X^{(r)}$ 为第 $r+1$ 轮 L 层的输入状态 (即该轮 S 层输出、 L 层输入的 1024 比特状态), 其左/右半部分分别记为 $X_L^{(r)}, X_R^{(r)}$ (各 512 比特). 令 $r_m - 1$ 表示匹配点所在轮的 L 输入索引.

• $B_L \triangleq X_L^{(r_m-1)}$ 的“前向版本”: 它由前向块在起点 r_s 处枚举蓝色中性半字节 (变量向量 v_B , 见算法 1 第 3 行) 得到的候选状态在 r_m 处的左半部. 实现上, 算法 1 的“表 B ”逐个存入这些 $X_L^{(r_m-1)}$, 因此当 $Rounds = 13$ 时, 文中记号 $B_L^{(12)}$ 即表示 $r_m - 1 = 12$ 时的该左半部候选.

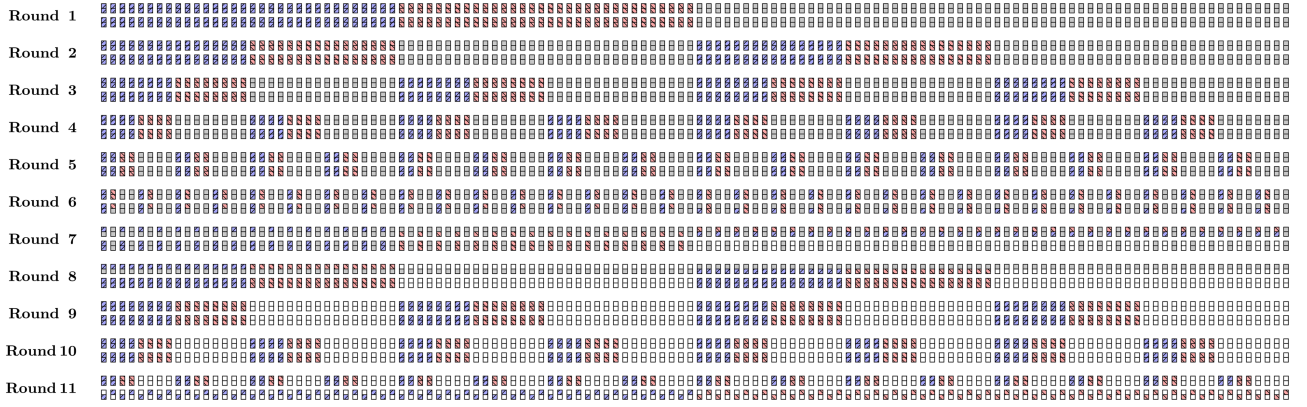


图 4 (网络版彩图) 11 轮 JH-512 的中间相遇特征.

Figure 4 (Color online) 11-round MitM characteristic of JH-512.

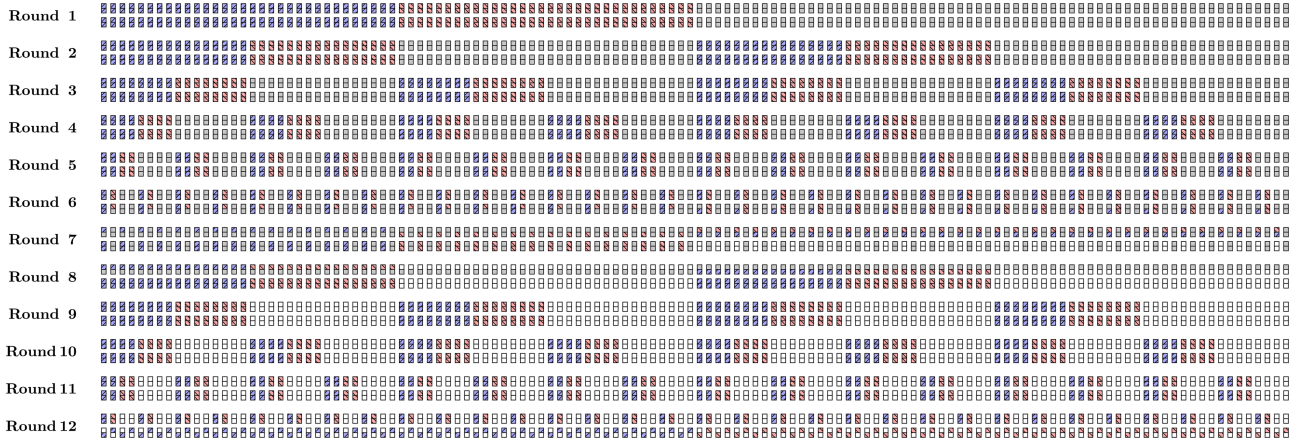


图 5 (网络版彩图) 12 轮 JH-512 的中间相遇特征.

Figure 5 (Color online) 12-round MitM characteristic of JH-512.

• $R_L \triangleq X_L^{(r_m-1)}$ 的“后向版本”: 它由后向块在起点 r_s 处枚举红色中性半字节 (变量向量 v_R , 见算法 1 第 7 行), 并经由 (2) 式的关系回溯, 再按 P_d^{-1} 对齐到匹配点后得到的候选状态在 r_m 处的左半部. 实现上, 算法 1 的“表 R ”逐个存入这些 $X_L^{(r_m-1)}$, 因此当 $Rounds = 13$ 时, 文中记号 $R_L^{(12)}$ 即表示 $r_m - 1 = 12$ 时的该左半部候选.

由图 3 可知, 根据 (18) 式, 中间状态应满足 64 个半字节条件:

$$\begin{cases} L(B_L^{(12)}) = P_d^{-1}(H_R^{(i)} \oplus A_L^{(0)} \oplus H_L^{(i-1)}), \\ L(R_L^{(12)}) = P_d^{-1}(H_R^{(i)} \oplus A_L^{(0)} \oplus H_L^{(i-1)}). \end{cases} \quad (18)$$

13 轮 JH-512 原像攻击流程. 基于 JH-512 的 13 轮 MitM 特征, 我们对 13 轮 JH-512 实施中间相遇原像攻击. 算法 1 给出了攻击流程.

针对 JH-512 杂凑函数的 13 轮原像攻击中, $n = 128$, $\min\{DoB_T, DoR_T, DoM_T\} = 48$, 因此总复杂度为 $2^{4 \times (n - \min\{DoB_T, DoR_T, DoM_T\})} = 2^{4 \times (128 - 48)} = 2^{320}$.

4.2 11 轮、12 轮 JH-512 算法原像攻击

在相同的建模与求解设置下, 得到如下的两组 MitM 特征.

11 轮特征. 如图 4 所示, 其中 $n = 128$, $DoB_T = DoR_T = 48$, $DoM_T = 64$, $\min\{DoB_T, DoR_T, DoM_T\} = 48$. 对应的针对 11 轮 JH-512 算法的原像攻击时间复杂度为 $2^{4 \times (n - \min\{DoB_T, DoR_T, DoM_T\})} = 2^{4 \times (128 - 48)} = 2^{320}$.

12 轮特征. 如图 5 所示, 其中 $n = 128$, $DoB_T = DoR_T = 48$, $DoM_T = 64$, $\min\{DoB_T, DoR_T, DoM_T\} = 48$. 对应的针对 12 轮 JH-512 算法的原像攻击时间复杂度为 $2^{4 \times (n - \min\{DoB_T, DoR_T, DoM_T\})} = 2^{4 \times (128 - 48)} = 2^{320}$.

上述两项结果均在更长轮数上优于既有工作的时间复杂度水平.

5 结论

本文提出面向 JH 杂凑函数的 MitM 建模与分析框架, 结合 MILP 自动搜索给出了 JH-512 的首个 11, 12 和 13 轮原像攻击结果, 并在 13 轮上实现时间复杂度 2^{320} 的改进结果, 建立了从建模到攻击的可复用范式. 该方法更新了针对 JH-512 缩减轮数版本的最佳已知攻击结果, 并为 AES 类与 Even-Mansour 类原语的安全评估提供了一种普适性的方法论. 后续可结合 MILP+SAT 的混合求解与分层/部分匹配策略, 推动更深轮数特征与更优时间-空间折中, 并将框架推广至 JH-224/256/384 及同类原语, 形成可复现实验基准并提炼可量化的抗 MitM 攻击设计准则.

参考文献

- Diffie W, Hellman M E. Special feature exhaustive cryptanalysis of the NBS data encryption standard. *Computer*, 1977, 10: 74–84
- van Oorschot P C, Wiener M J. A known plaintext attack on two-key triple encryption. In: *Proceedings of Advances in Cryptology—EUROCRYPT'90*, Aarhus, 1990. 318–325
- Merkle R C, Hellman M E. On the security of multiple encryption. *Commun ACM*, 1981, 24: 465–467
- Dunkelman O, Sekar G, Preneel B. Improved meet-in-the-middle attacks on reduced-round des. In: *Proceedings of Progress in Cryptology—INDOCRYPT 2007*, Chennai, 2007. 86–100
- Aoki K, Sasaki Y. Preimage attacks on one-block md4, 63-step md5 and more. In: *Proceedings of Selected Areas in Cryptography*, Sackville, 2008. 103–119
- Sasaki Y, Aoki K. Finding preimages in full md5 faster than exhaustive search. In: *Proceedings of Advances in Cryptology—EUROCRYPT 2009*, Cologne, 2009. 134–152
- Bogdanov A, Khovratovich D, Rechberger C. Biclique cryptanalysis of the full aes. In: *Proceedings of Advances in Cryptology—ASIACRYPT 2011*, Seoul, 2011. 344–371
- Khovratovich D, Leurent G, Rechberger C. Narrow-bicliques: cryptanalysis of full idea. In: *Proceedings of Advances in Cryptology—EUROCRYPT 2012*, Cambridge, 2012. 392–410
- Bogdanov A, Rechberger C. A 3-subset meet-in-the-middle attack: cryptanalysis of the lightweight block cipher ktantan. In: *Proceedings of Selected Areas in Cryptography—SAC 2010*, Waterloo, 2010. 229–240
- Canteaut A, Naya-Plasencia M, Vayssière B. Sieve-in-the-middle: improved mitm attacks. In: *Proceedings of Advances in Cryptology—CRYPTO 2013*, Santa Barbara, 2013. 222–240
- Fuhr T, Minaud B. Match box meet-in-the-middle attack against katan. In: *Proceedings of Fast Software Encryption—FSE 2014*, London, 2014. 61–81
- Dinur I, Dunkelman O, Keller N, et al. Efficient dissection of composite problems, with applications to cryptanalysis, knapsacks, and combinatorial search problems. In: *Proceedings of Advances in Cryptology—CRYPTO 2012*, Santa Barbara, 2012. 719–740
- Demirci H, Selçuk A A. A meet-in-the-middle attack on 8-round aes. In: *Proceedings of Fast Software Encryption—FSE 2008*, Lausanne, 2008. 116–126
- Dunkelman O, Keller N, Shamir A. Improved single-key attacks on 8-round aes-192 and aes-256. In: *Proceedings of Advances in Cryptology—ASIACRYPT 2010*, Singapore, 2010. 158–176
- Li L B, Jia K T, Wang X Y. Improved single-key attacks on 9-round aes-192/256. In: *Proceedings of Fast Software Encryption—FSE 2014*, London, 2014. 127–146
- Gilbert H, Minier M. A collision attack on 7 rounds of rijndael. In: *Proceedings of the Third Advanced Encryption Standard Candidate Conference*, New York, 2000. 230–241
- Wei L, Rechberger C, Guo J, et al. Improved meet-in-the-middle cryptanalysis of ktantan (poster). In: *Proceedings of Information Security and Privacy—ACISP 2011*, Melbourne, 2011. 433–438
- Liu Y, Cheng L, Liu Z Q, et al. Improved meet-in-the-middle attacks on reduced-round Piccolo. *Sci Sin Inform*, 2018, 61: 032108 [刘雅, 程亮, 刘志强, 等. 改进的针对缩减轮 Piccolo 的中间相遇攻击. *中国科学: 信息科学*, 2018, 61: 032108]
- Isobe T, Shibutani K. All subkeys recovery attack on block ciphers: extending meet-in-the-middle approach. In: *Proceedings of Selected Areas in Cryptography—SAC 2012*, Windsor, 2012. 202–221

- 20 Knellwolf S, Khovratovich D. New preimage attacks against reduced sha-1. In: *Proceedings of Advances in Cryptology—CRYPTO 2012*, Santa Barbara, 2012. 367–383
- 21 Espitau T, Fouque P A, Karpman P. Higher-order differential meet-in-the-middle preimage attacks on sha-1 and blake. In: *Proceedings of Advances in Cryptology—CRYPTO 2015*, Santa Barbara, 2015. 683–701
- 22 Boura C, David N, Derbez P, et al. Differential meet-in-the-middle cryptanalysis. In: *Proceedings of Advances in Cryptology—CRYPTO 2023*, Santa Barbara, 2023. 240–272
- 23 Song L, Yang Q Q, Liu H M. Revisiting the differential meet-in-the-middle cryptanalysis. 2023. <https://eprint.iacr.org/2023/1302>
- 24 Ahmadian Z, Khalesi A, M'foukh D, et al. Improved differential meet-in-the-middle cryptanalysis. In: *Proceedings of Advances in Cryptology—EUROCRYPT 2024*, Zurich, 2024. 280–309
- 25 Grochal P, Stanek M. Differential MITM attacks on SLIM and LBCIoT. 2024. <https://eprint.iacr.org/2024/1887>
- 26 Liu D, Luo Y B, Jia K T, et al. Differential analysis of block cipher FBC. *Sci Sin Inform*, 2024, 54: 335–353 [刘端, 罗毅博, 贾珂婷, 等. 分组密码 FBC 的差分分析. *中国科学: 信息科学*, 2024, 54: 335–353]
- 27 Chen S W, Hu K, Liu G Z, et al. Meet-in-the-middle attack on 4+4 rounds of SCARF under single-tweak setting. 2024. <https://eprint.iacr.org/2024/1270>
- 28 Zhang P, Luo Y Y. Beyond-birthday-bound (BBB) secure tweakable cryptographic primitives based on tweakable Even-Mansour cipher instances. *Sci Sin Inform*, 2025, 55: 1007–1032 [张平, 罗宜元. 基于可调 Even-Mansour 密码实例构造超生日界安全的可调密码本原. *中国科学: 信息科学*, 2025, 55: 1007–1032]
- 29 Bouillaguet C, Derbez P, Fouque P A. Automatic search of attacks on round-reduced aes and applications. In: *Proceedings of Advances in Cryptology—CRYPTO 2011*, Santa Barbara, 2011. 169–187
- 30 Derbez P, Fouque P A. Automatic search of meet-in-the-middle and impossible differential attacks. In: *Proceedings of Advances in Cryptology—CRYPTO 2016*, Santa Barbara, 2016. 157–184
- 31 Shi D P, Sun S W, Derbez P, et al. Programming the demirci-selçuk meet-in-the-middle attack with constraints. In: *Proceedings of Advances in Cryptology—ASIACRYPT 2018*, Brisbane, 2018. 3–34
- 32 Sasaki Y. Integer linear programming for three-subset meet-in-the-middle attacks: application to gift. In: *Proceedings of Advances in Information and Computer Security—IWSEC 2018*, Sendai, 2018. 227–243
- 33 Bao Z Z, Dong X Y, Guo J, et al. Automatic search of meet-in-the-middle preimage attacks on aes-like hashing. In: *Proceedings of Advances in Cryptology—EUROCRYPT 2021*, Zagreb, 2021. 771–804
- 34 Dong X Y, Hua J L, Sun S W, et al. Meet-in-the-middle attacks revisited: key-recovery, collision, and preimage attacks. In: *Proceedings of Advances in Cryptology—CRYPTO 2021*, 2021. 278–308
- 35 Bao Z Z, Guo J, Shi D P, et al. Superposition meet-in-the-middle attacks: updates on fundamental security of aes-like hashing. In: *Proceedings of Advances in Cryptology—CRYPTO 2022*, Santa Barbara, 2022. 64–93
- 36 Schrottenloher A, Stevens M. Simplified mitm modeling for permutations: new (quantum) attacks. In: *Proceedings of Advances in Cryptology—CRYPTO 2022*, Santa Barbara, 2022. 717–747
- 37 Schrottenloher A, Stevens M. Simplified modeling of MITM attacks for block ciphers: new (quantum) attacks. *IACR Trans Symmetric Cryptol*, 2023, 2023: 146–183
- 38 Shi D P, Sun S W, Song L, et al. Exploiting non-full key additions: full-fledged automatic demirci-selçuk meet-in-the-middle cryptanalysis of skinny. In: *Proceedings of Advances in Cryptology—EUROCRYPT 2023*, Lyon, 2023. 67–97
- 39 Hou Q L, Dong X Y, Qin L Y, et al. Automated meet-in-the-middle attack goes to feistel. In: *Proceedings of Advances in Cryptology—ASIACRYPT 2023*, Guangzhou, 2023. 370–404
- 40 Hou Q L, Li K T, Zhang G Y, et al. Meet-in-the-middle attack on primitives with binary matrix linear layer. In: *Proceedings of Topics in Cryptology—CT-RSA 2025*, San Francisco, 2025. 51–74
- 41 Vaziri M. Automated tool for meet-in-the-middle attacks with very low data and memory complexity (full version). 2025. <https://eprint.iacr.org/2025/1401>
- 42 Chakraborty D, Sahoo S, Nguyen P H, et al. An automated model to search for differential meet-in-the-middle attack: applications to andrx ciphers. 2025. <https://eprint.iacr.org/2025/1249>
- 43 He J H, Hu K, Lei H, et al. Massive superpoly recovery with a meet-in-the-middle framework. In: *Proceedings of Advances in Cryptology—EUROCRYPT 2024*, Zurich, 2024. 368–397
- 44 Dong X Y, Zhao B X, Qin L Y, et al. Generic mitm attack frameworks on sponge constructions. In: *Proceedings of Advances in Cryptology—CRYPTO 2024*, Santa Barbara, 2024. 3–37
- 45 Gohr A. Improving attacks on round-reduced speck32/64 using deep learning. In: *Proceedings of Advances in Cryptology—CRYPTO 2019*, Santa Barbara, 2019. 150–179
- 46 Shen Y T, Chen Y, Yu H B. General enhancing framework for deep learning-aided cryptanalysis: applications to Speck, Simon and LEA ciphers. *Sci Sin Inform*, 2025, 55: 1447–1470 [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: 1447–1470]

- 47 Carlini N, Chávez-Saab J, Hambitzer A, et al. Polynomial time cryptanalytic extraction of deep neural networks in the hard-label setting. In: Proceedings of Advances in Cryptology—EUROCRYPT 2025, Madrid, 2025. 364–396
- 48 Gérault D, Hambitzer A, Ronen E, et al. How to securely implement cryptography in deep neural networks. 2025. <https://eprint.iacr.org/2025/288>
- 49 Zhang Z Y, Sun S W, Wang C B, et al. Classical and quantum meet-in-the-middle nostradamus attacks on aes-like hashing. IACR Trans Symmetric Cryptol, 2023, 2023: 224–252
- 50 Zhang T Y. Comprehensive preimage security evaluations on rijndael-based hashing. In: Proceedings of Applied Cryptography and Network Security Workshops—ACNS 2023, Kyoto, 2023. 23–42
- 51 Chen S Y, Guo J, List E, et al. Diving deep into the preimage security of aes-like hashing. In: Proceedings of Advances in Cryptology—EUROCRYPT 2024, Zurich, 2024. 398–426
- 52 Chen S Y, Guo J, List E, et al. Scrutinizing the security of AES-based hashing and one-way functions. 2025. <https://eprint.iacr.org/2025/792>
- 53 Wu H J. The hash function JH (SHA-3 Round 3 Submission). 2011. <https://www3.ntu.edu.sg/home/wuhj/research/jh/jh-round3.pdf>
- 54 Bhattacharyya R, Mandal A, Nandi M. Security analysis of the mode of JH hash function. In: Proceedings of Fast Software Encryption, Seoul, 2010. 168–191
- 55 Naya-Plasencia M, Toz D, Varici K. Rebound attack on JH42. In: Proceedings of Advances in Cryptology—ASIACRYPT 2011, Seoul, 2011. 252–269
- 56 Nourizadeh S, Javanmardi M, Sadeghiyan B. Cryptanalysis of the reduced-round version of JH. In: Proceedings of the 6th International Symposium on Telecommunications, IST 2012, Tehran, 2012. 1050–1054
- 57 Luo Y Y, Lai X J. Attacks on JH, grøstl and smash hash functions. 2013. <https://eprint.iacr.org/2013/233.pdf>
- 58 Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2024. <https://www.gurobi.com>
- 59 Sasaki Y. Meet-in-the-middle preimage attacks on AES hashing modes and an application to whirlpool. In: Proceedings of Fast Software Encryption—FSE 2011, Lyngby, 2011. 378–396
- 60 Sun S W, Hu L, Wang P, et al. Automatic security evaluation and (related-key) differential characteristic search: application to simon, present, lblock, DES(L) and other bit-oriented block ciphers. In: Proceedings of Advances in Cryptology—ASIACRYPT 2014, Kaoshiung, 2014. 158–178

MILP-based meet-in-the-middle preimage attack on the JH-512 hash function: breaking 13 rounds

Zheng LI^{1,2}, Yimin HE¹, Hailun YAN^{3,4*}, Liang ZHAO² & Jianbiao ZHANG¹

1. College of Computer Science, Beijing University of Technology, Beijing 100124, China

2. Key Laboratory of Data Protection and Intelligent Management, Ministry of Education, Sichuan University, Chengdu 610065, China

3. School of Cryptology, University of Chinese Academy of Sciences, Beijing 100049, China

4. State Key Laboratory of Cryptology, Beijing 100878, China

* Corresponding author. E-mail: hailun.yan@ucas.ac.cn

Abstract Recently, meet-in-the-middle (MitM) attacks have achieved notable progress in the cryptanalysis of both advanced encryption standard (AES)-like and sponge-based constructions. As a SHA-3 competition finalist, the JH hash function adopts a permutation-based compression function related to the Even-Mansour structure; its underlying permutation is designed using a generalized AES design methodology. This paper proposes a MitM modeling framework tailored to the JH-512 permutation and employs mixed-integer linear programming (MILP) to automatically search for a 13-round MitM characteristic. Building on this, we mount a MitM preimage attack on JH-512 reduced to 13 out of 42 rounds, with a time complexity of 2^{320} , improving upon the previous best known preimage attack on 10 out of 42 rounds with a time complexity of 2^{325} . While the full-round JH hash function remains secure, our approach provides new insights and techniques for the cryptanalysis of JH and other related primitives.

Keywords hash function, Even-Mansour construction, meet-in-the-middle attack, preimage attack, JH