基于神经网络的输出预测攻击对对称密钥密码设计的影响*

渡边隼[1]、伊藤龙马[2[0000-0002-4929-8974]]和大石俊弘[1 2]

[1] 日本东京都港区德岛大学。
3CJNM024@tokai.ac.jp,ohigashi@tokai.ac.jp
[2] 日本Koganei NICT。
itorym@nict.go.jp

摘要。要设计安全的对称密钥密码，必须证明其对传统攻击（如差分攻击、线性攻击和积分攻击）的抵抗能力。尽管自动搜索和深度学习方法的进步使这项耗时的任务变得相对容易，但专家需求和潜在疏漏的问题仍令人担忧。为解决这些问题，Kimura等人提出了基于神经网络的输出预测（NN）攻击，该方法具有操作简便、通用性强且编码错误率低等优势。这种攻击方式对设计安全的对称密钥密码尤其有益，特别是基于S盒的分组密码。受其研究启发，我们首先将NN攻击应用于ANDRotation-XOR架构的分组密码Simon，识别出易受攻击的结构及其检测到的漏洞。随后深入分析这些脆弱结构，发现其中最易受攻击的结构在扩散特性方面表现最弱。这一事实表明，神经网络攻击可能能够检测到此类特性。我们随后聚焦于易受攻击的西蒙类密码核心函数中的偏差事件，并构建由该事件引发的有效线性近似模型。最终，通过这些线性近似模型揭示：相较于原始结构，易受攻击的密码结构更容易遭受线性密钥恢复攻击。
我们得出的结论是，我们的分析可以成为使NN攻击成为设计安全对称密钥密码的有用工具的一个坚实步骤。

关键词：分组密码·西蒙算法·设计原理·神经网络·输出预测攻击

1 介绍

在设计新型对称密钥密码时，最关键的评估指标之一就是验证其是否具备抵御通用攻击的能力。这类攻击包括差分攻击[8]、线性攻击[24]、不可能差分攻击[7,19]、线性不可约攻击（即zerocorrelation_linear[9]）以及积分攻击（即[20]）。通常来说，

* 本文的部分内容在第八届国际网络安全、密码学和机器学习研讨会（CSCML 2024）上发表。

由线性和非线性运算的组合构成，每个运算都有许多设计选择（例如，旋转参数）。这意味着设计者必须评估所有可能从这些选择构建的候选密码的安全性；因此，设计一种新的密码可能会非常耗时。

基于混合整数线性规划（MILP）、布尔可满足性问题（SAT）和约束编程（CP）的自动搜索方法[15,16,26-29]，在过去十年间迅速发展成为辅助工具，使这项耗时的任务变得相对容易。然而，使用这些工具存在以下问题：1）需要具备分析和建模方法的高级专业知识；2）建模方法因攻击向量不同而有所差异；3）由于编码错误可能导致未知漏洞被忽视。实际上，已有多个案例表明，在新设计规范发布后不久，就发现了设计者未曾预料到的未知漏洞。因此，目标密码随后遭到破解。例如，针对SPEDDY [10]的差分攻击、Friet [30]的差分攻击以及Chaghri [23]的代数攻击。除了自动搜索方法外，基于深度学习的分析方法在过去五年间也得到了快速发展。据我们所知，这些方法从未被用于设计新型对称密钥密码。因此，本研究旨在探讨将这些方法应用于此类场景的可能性。

自戈尔[14]在2019年CRYPTO密码学会议上提出新型差分神经密码分析方法以来，基于深度学习的对称密钥密码分析技术已取得显著进展，例如[2,3,5,6]算法。然而现有方法大多采用差分攻击与深度学习技术相结合的方式，这可能导致其自动搜索机制存在与传统方法相同的三大缺陷。为解决这些问题，木村等人[17,18]提出了一种名为基于神经网络的输出预测（NN）攻击的新方法，该方法具有以下三大特性：1）无需具备高级分析建模专业知识[1]；2）能够应对所有攻击向量；3）由于仅需输入/输出接口参数（如分组大小）作为输入，因此编码错误率较低。研究团队将NN攻击应用于基于S盒的分组密码，最终得出结论：NN攻击有望成为设计新型对称密钥密码（尤其是基于S盒的密码）的有效工具。

本研究旨在推动神经网络攻击技术发展，使其成为设计各类对称密钥密码的有力工具。为实现这一目标，我们以美国国家安全局（NSA）[4]开发的AND-旋转-XOR结构分组密码Simon为例，将其作为神经网络攻击的研究对象。在设计类似Simon的密码方案时，根据三个旋转参数的不同组合，可考虑多种候选设计方案。，（a、b、c）；然后，通过改变旋转参数，我们试图获得新的见解，以考虑对称密钥密码的设计原理，特别是西蒙式密码。

我们的研究受到Kolbl等人[21,22]和Kondo等人[21,22]工作的启发，我们旨在从他们的工作角度回答以下问题：

[1] 换句话说，它使我们能够用神经网络的基本知识进行编码。

1. 在西蒙式密码中，哪些结构容易受到NN攻击?
2. NN攻击检测到哪些类型的漏洞?
3. 如果检测到漏洞，我们能否找出其根本原因?
4. 脆弱的密码对密钥恢复攻击有多大的抵抗力?

解开这些问题将是使NN攻击成为设计一种新的对称密钥密码的有用工具的一个坚实步骤。

我们的贡献总结如下:

识别易受攻击的结构。在第3节中，我们解答了第一个问题。具体而言，我们将神经网络攻击成功的最大轮数与构建差分/线性区分器所需的最大学轮数进行对比。随后，我们明确了在西蒙类密码中，哪些结构比差分和线性攻击更容易遭受神经网络攻击。实验结果表明，包含以下两种情况之一的西蒙类密码存在未知漏洞：1）“a=c”或“b=c”；2）“n-a=c”或“n-b=c”（其中n表示块大小）。

脆弱结构的综合分析。见第3节。第三部分，我们基于上述研究成果对第二个问题进行深入探讨。具体而言，针对易受攻击的西蒙类密码，我们开展了补充实验以更全面地分析其特性，并探讨神经网络攻击能检测到哪些类型的漏洞。这些漏洞不仅包括差分攻击和线性攻击，还涉及不可能差分攻击、零相关线性攻击以及积分攻击。研究结果表明，神经网络攻击有可能检测到由积分攻击引发的漏洞，而包含“a=c”或“b=c”以及‘0’或‘n/2’参数的西蒙类密码结构最为脆弱。

探究易受攻击结构的根本原因。在第四节中，我们将从以下两个方面解答第三个问题。首先针对广为人知的扩散特性，我们证明了最易受攻击的西蒙类密码相较于其他类型具有最低的扩散性能。其次分析目标密码核心函数输出值存在偏差的影响，这种偏差输出事件使我们能够找到有效的线性近似方法。根据我们的发现，可以为最脆弱的西蒙式密码构建最多30轮的线性鉴别器。

针对易受攻击的西蒙类密码进行密钥恢复研究。在第五节中，我们为第四个问题提供了可能的解答方案。基于已知的松井算法1 [24]，我们运用发现的线性近似方法实施了线性密钥恢复攻击。实验结果表明，相较于原始密码，易受攻击的西蒙类密码更容易遭受此类密钥恢复攻击。此外，我们还发现了一个新现象：随着样本数量增加，攻击能力并未随之提升。解决这一问题将成为我们未来的研究重点。

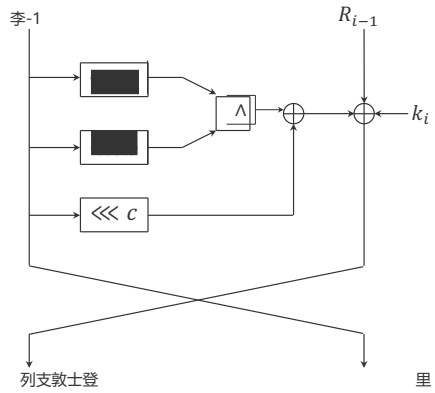


图1：Simon的圆形函数。

2 预备会议

2.1 西蒙规格

由NSA于2013年[4]设计的西蒙（Simon）系列，是基于Feistel构造的轻量级分组密码算法。该系列包含十个变体，每个变体结合2n位分组和mn位密钥，其中n∈{16,24,32,48,64}，m∈{2,3,4}。虽然西蒙变体通常表示为Simon2n/mn，但本文为简化表述统一使用Simon2n，因密钥长度不在本研究分析范围内。本研究主要针对具有32轮运算的Simon32进行探讨。此外，为验证分析有效性，我们特别设计了Simon16作为西蒙系列的简化模型。

如图1所示，西蒙的轮函数由三个n位运算组成：与（∧）、左移（。）和异或（ ）。设x是轮函数中核心函数f的n位输入，则该函数可定义为

函数f(x)定义为：当x。a且x。b时，其值为（x。a） ，否则为（x²）c。 (1)

其中（a，b，c）表示旋转参数，（a，b，c）=（1,8,2）用于原始Simon32。为简化表述，本文将所有可能旋转参数的Simon2n称为Simon2n变体[2]。此外，设（Li_1，Ri_1）为第i轮函数的2n位输入；则其输出（Li，Ri）的计算方式为

锂的反应式为Li = f（Li_1） Ri_1 ki， (2)

$$R_i = L_{i-1}, \tag{3}$$

其中ki表示第i轮的子密钥。对于r轮西蒙算法，其参数为plaintext__is__（L_0，R_0），密文为（L_r，R_r）。

[2] 注意，这与Simon变体的符号不同，后者包括Simon32、Simon48、Simon64、Simon96和Simon128的原始版本。

2.2 关于Simon变体设计原理的考虑

科尔布、莱安德和蒂森[21]从三个维度探讨了西蒙变体的设计原理。首先，他们测量了所有可能旋转参数下各西蒙变体达到完全扩散所需的轮数。接着，针对满足a > b且（a-b，n）最大公约数为1的Simon32、Simon48和Simon64变体，计算了10轮时的最佳微分/线性特性参数，并列出了20个最优参数（详见[21]附录C）。最后，通过进一步评估，他们针对以下三个示例参数：（12,5,3）、（7,0,2）和（1,0,2），获得了Simon32、Simon48和Simon64变体的微分特性。研究最终表明，原始旋转参数未必总是最优选择。

近藤、佐佐木、Todo和Iwata [22]专注于Simon32的设计原理，并扩展了Kolbl等人的工作[21]，以针对Simon32变体的积分和不可能的差分攻击。针对积分攻击，研究团队利用超级计算机评估了使用2^{31}个明文构建积分区分器所需的轮数。对于不可能差分攻击，他们采用通用计算机通过中间缺失法探索构建不可能差分区分器所需的轮数。基于这些研究成果，研究人员将Simon32变体划分为20组（详见[22]中的表6），并明确了在对抗积分攻击和不可能差分攻击时，Simon32算法中哪个参数最为优化。

2.3 基于深度学习的输出预测攻击

Kimura等人[17] 研究者在黑盒环境下提出了基于深度学习的输出预测（DL-OP）攻击方法，并首次将其应用于三个16位块大小的玩具级分组密码：两个SPN分组密码（小型PRESENT和小型AES)以及一个Feistel分组密码（小型TWINE)。随后，基于对这些玩具级分组密码的分析结果，他们还将该攻击方法扩展至三个32位和64位块大小的分组密码：PRESENT、类AES型和类TWINE型密码。Kimura等人研究团队提出了一种基于深度学习的输出预测攻击方法，采用长短期记忆网络（LSTM）作为神经网络模型，并将其应用于块密码算法PRESENT、AES和TWINE的玩具模型。结论表明：对于PRESENT算法，他们证明该攻击等效于构建差分和线性区分器；而对于类似AES和TWINE的密码算法，当训练数据量进一步增加时，推测该攻击同样等效于构建差分和线性区分器。

在后续研究[18]中，研究者们将先前工作[17]的成果进一步拓展，旨在探索设计能抵御差分线性攻击（DLOP）的对称密钥密码的线索。为此，他们采用两种弱小型PRESENT变体方案——通过用已知易受差分攻击和线性攻击的弱S盒替换原始S盒，并沿用前期研究的方法对这些变体实施DLOP攻击。最终，他们证明了

DLOP攻击在估算构建差分和/或线性区分器所需的轮数方面效果显著。然而，如何通过其结果为设计抗差分勒索攻击的对称密钥密码提供反馈机制，目前仍不明确。

2.4 基于SAT/CP的自动搜索方法

我们拓展了现有研究[17,18,21,22]，并从神经网络攻击的视角深入探讨Simon32的设计原理。与Kolbl等人[21,22]和Kondo等人[21,22]的研究不同，我们的分析明确了Simon32变体中哪个参数是非最优的；这将有助于我们在设计类Simon密码时判断哪些参数不应被选用。

我们还明确了神经网络攻击能检测哪些漏洞。为此，我们采用了现有的基于SAT/CP的自动搜索工具。值得庆幸的是，许多开源工具可用来通过通用攻击手段（如差分攻击、线性攻击、不可能差分攻击、零相关线性攻击和积分攻击）来查找对称密钥密码的各种区分器。随后，我们定制并使用了以下三种工具来分析满足a > b且（a-b，n）=1[3]条件的Simon32变体：

- 为了评估构建差分和线性区分器的最大轮数，我们使用了Sun等人在IACR ToSC 2021会议上提出的基于SAT的工具(1)[29]。源代码可在GitHub[4]上获取。
- 为了评估构建不可能差分器和零相关线性区分器的最大轮数，我们采用了Hadipour等人在IACR ToSC 2024(1) [16]会议上提出的基于CP的工具。源代码可在GitHub5获取。
- 为了评估构建积分区分器的最大轮数，我们采用了Hadipour等人在IACR ToSC 2022(2) [15]会议上提出的基于SAT的工具。源代码可在GitHub[6]上获取。

2.5 复杂性估计和成功概率

我们将神经网络攻击视为一种区分性攻击，换言之，如果神经网络攻击能以高于随机预测的概率预测出某个输出比特串，则认为该攻击成功。

为了估计样本数量和成功概率，以区分两个分布相对于输出比特串的情况，我们使用了Baigneres等人在ASIACRYPT 2004 [1]会议上提供的以下定理。

定理1（[1，定理6]）。设Z1，...，Z_n 是分布D_0 和D_1 上的独立同分布随机变量，且D_0 和D_1 为两个

[3] 必须明确这些条件，以便使用Sun等人的方法[29]来寻找差异特征。

[4] https://github.com/SunLing134340/Accelerating_Automatic_Search
[5] https://github.com/hadipourh/zeroplus
[6] https://github.com/hadipourh/mpt

具有相同支撑集且彼此接近的分布，其中n表示在D= D_0 或D = D_1 中最佳区分器的样本数量。设d为一个实数，使得

$$n = \frac{d}{\sum_{z \in \mathcal{Z}} \frac{\epsilon_z^2}{p_z}}, \tag{4}$$

其中pz和pz+ωz分别表示随机变量z在D_0 和D_1 分布下的概率。那么，总体误差概率为P_e ≈ j （-√d/2），其中 j （·）是标准正态分布的分布函数。

设D_0 和D_1 分别为随机预测结果的分布和神经网络攻击结果的分布。在此情况下，目标事件发生于D_0 和D中。with probabilities of $\frac{1}{2}$ and $\frac{1}{2}$ · (1 +ϵ)，分别（即，p_0 = p_1 =$\frac{1}{2}$，|ω_0 |=0，且|ω_1 >|=$\frac{\epsilon}{2}$)基于此，D=D_0 和D=D_1 之间最佳区分器的样本数量可估计为2dϵ^{-2}，总体错误概率为P_e ≈ j （—√d/2）；因此，成功概率可估计为1 — Pe。

3 关于NN攻击对Simon32变体的影响

在本节中，我们针对对称密钥密码（如差分攻击、线性攻击、不可能差分攻击、零相关线性攻击和积分攻击）进行了神经网络攻击及五种通用攻击的实验，并探讨了类西蒙密码的设计原理。首先，我们采用第2.4节所述基于SAT/CP的自动搜索方法对Simon32变体进行分析，推导出每种通用攻击所能构建区分器的最大轮数。随后，我们重点研究差分攻击和线性攻击的实验结果，并根据各变体构建区分器的最大轮数将其划分为四组。我们随机选取每组32个Simon32变体，应用第2.3节所述的神经网络攻击进行测试。通过对比神经网络攻击与差分、线性攻击的实验结果，我们明确了哪种Simon32变体对神经网络攻击比对差分和线性攻击更具脆弱性。换言之，这表明神经网络攻击可能检测到其他非差分和线性攻击所引发的漏洞。最后，为了深入研究存在参数漏洞的Simon32变体，我们针对这些变体进行了额外实验，并探讨了神经网络攻击能检测到哪些类型的漏洞。这些漏洞不仅包括差分攻击和线性攻击，还涉及不可能差分攻击、零相关线性攻击以及积分攻击。这表明神经网络攻击可以成为设计对称密钥密码（尤其是类Simon密码）的有效工具。

3.1 重新审视对Simon32变体的通用攻击

如第2.2节所述，Kolbl等人[21] 从差分攻击和线性攻击的角度考虑了西蒙变体的设计原理。受启发

通过他们的工作，近藤等人[22] 从积分攻击和不可能差分攻击的角度对Simon 32变体进行了深入研究。虽然可以使用他们的实验结果，但并非所有结果都公开可用，因此我们不得不自行获取实验数据。

我们采用第2.4节所述的现有基于SAT/CP的自动搜索工具，对Simon32变体针对对称密钥密码的五种通用攻击（差分攻击、线性攻击、不可能差分攻击、零相关线性攻击和积分攻击）的安全性进行全面分析。实验结果概览如下：

-可以用2^{15} 数据[7] 构建差分和线性区分器的最大轮数范围是7到15轮。
-能够构建不可能的差分和零相关线性区分器的最大轮数范围是9到17轮。
-可以用2^{31} 个数据建立一个积分鉴别器的最大轮数范围是14到32轮。

如本节开头所述，我们首先重点分析差分攻击和线性攻击的实验结果，并根据这些攻击测试结果将Simon32变体划分为四个类别。具体而言，我们将能够构建差分和线性区分器的最大轮数分别为15轮、11轮、8轮和7轮的Simon32变体分别归入A组、B组、C组和D组。

3.2 针对Simon32变体的NN攻击

在本小节中，我们为前文分类的每个组别随机选取了32个Simon32变体，并对这些变体实施神经网络攻击（特别是明文预测攻击）。首先，我们将阐述实验分析流程及成功攻击的条件。接着展示针对目标变体的神经网络攻击实验结果。最后，为验证实验结果的正确性，我们沿用相同方法对作为玩具密码设计的Simon16进行了神经网络攻击实验。

实验流程。我们基本遵循Kimura等人论文[17]第3.1节中阐述的实验流程，与他们方法的主要差异如下：

我们未进行超参数优化，因为难以确定所有目标变体的最佳超参数。取而代之的是，我们采用了Kimura等人论文[17]中表3所列的四轮小型PRESENT攻击方案中使用的超参数。这是因为针对小型PRESENT的神经网络攻击准确率

[7] 数据复杂度限制为2^{15} 的原因是，用于NN攻击的样本数量为2^{15}，我们主要比较了差分攻击和线性攻击与NN攻击的能力。

表1：实验超参数。

训练数据数量[t]	2^{15}
测试数据数量[t]	2^{15}
方法	LSTM（长短期记忆）
输入层节点数（即块大小）	16, 32
输出层节点数（即块大小）	16, 32
隐藏节点数	300
隐藏层数	1
损失函数	均方误差
优化程序	亚当
学习速率的初始值	0.01
批量大小	250
轮数	100

[t] 明文/密文对的数量（训练和测试数据中没有重复项）。

在目标密码中表现最佳，针对小型PRESENT的明文预测攻击最多成功四轮。表1列出了实验中使用的超参数设置。

——我们选择使用10个随机密钥而非100个，主要是考虑到实验执行时间的限制。有趣的是，Kimura等人在论文[17]第3.3节中证明，即使使用少量密钥进行实验，也能获得最佳平均成功率。因此我们认为，减少密钥数量对实验结果的影响微乎其微。

实验环境如下：四台Linux主机，配备八个NVIDIA Tesla K40M GPU。实验涵盖了1到16轮Simon32变体，每轮进行十次试验[8]。我们利用上述实验环境，在一天内完成了10个Simon32变体的实验结果获取。

成功实施神经网络攻击的条件。作为判断神经网络攻击是否成功的标准，Kimura等人[17] 我们定义了精确匹配率，即预测值与真实值在2n位输出字符串中完全一致的概率。具体而言，当数据块大小为2n位且训练数据量为2^x时，若精确匹配率超过（$2^{2n} - 2^x$）$^{-1}$则判定攻击成功。该条件适用于$2^{2n} = 2^x + 2^y$的情况，其中2^y代表测试数据量。例如，我们可在Simon16变体实验中应用此条件。但当满足该条件时，由于我们限制了$2^x = 2^y = 2^{15}$，即对于^{2n}、^{x}+^{y}的情况，我们无法精确计算出确切的匹配率；因此，我们无法严格判定对Simon32__变体的攻击是否成功。

[8] 如果NN攻击成功进行到16轮，我们继续实验，增加轮数直到攻击失败。

表2：针对Simon32变体（参数设置为4、1、12)的神经网络攻击实验结果。其中，DCP和LCP分别代表差分特征概率与线性特征概率。

Round	DCP	LCP	Average match rate	Exact match rate	success (✓) or failure (–)
1	1	1	0.99047	$2^{-0.44}$	✓
2	2^{-2}	2^{-2}	0.57558	$2^{-25.5}$	✓
3	2^{-4}	2^{-4}	0.55783	$2^{-26.9}$	✓
4	2^{-6}	2^{-6}	0.54309	$2^{-28.2}$	✓
5	2^{-8}	2^{-8}	0.52027	$2^{-30.2}$	✓
6	2^{-10}	2^{-10}	0.51634	$2^{-30.5}$	✓
7	2^{-12}	2^{-12}	0.51215	$2^{-30.9}$	✓
8	2^{-14}	2^{-14}	0.50619	$2^{-31.4}$	✓
9	2^{-18}	2^{-16}	0.50413	$2^{-31.6}$	–

为解决这一问题，我们定义了平均匹配率。具体来说，该指标的计算方法是：首先对每个比特位进行预测概率计算，然后将所有2^n个比特位的预测概率取平均值。例如，当平均匹配率为2^{-1}时，我们可以用2^{-32}作为精确匹配率的近似值——这在针对Simon32变体的神经网络攻击场景中尤为重要。

在此，我们遵循定理1并定义一个通过平均匹配率判断实验成败的条件。假设区分攻击的成功概率为0.7，根据定理可得d≈1.12。由于测试数据量为2^{15} =2 $d\epsilon^{-2}$，因此得到ϖ≈0. 00826；由此可推导出$2^{-1}\cdot(1+\varpi)$ ≈0. 504，这可以视为神经网络攻击结果的分布特征。这意味着当平均匹配率高于0.504时，NN攻击成功的概率可达0.7。综上所述，我们考虑一个较小的容差范围，并将平均匹配率达到0.505或更高时定义为成功攻击。

实验结果。我们针对Simon32变体开展了神经网络攻击实验，并将实验结果分为四组进行分类，具体如表3所示。该表格对比了各Simon32变体中构建差分/线性区分器的最大轮数（“D/L”列）与神经网络攻击成功所需的轮数（“NN”列），其中“组别”列定义详见第3.1节。以表2中（4,1,12）参数的Simon32变体为例，从差异特征概率和线性特征概率（参见“DCP”和“LCP”列）[9]来看，攻击在最多八轮内即可成功，这表明差异攻击、线性攻击与神经网络攻击的能力具有可比性。

[9] 严格来说，微分（分别。此处应考虑线性）概率，但由于基于DL的分析预计会检测到单一差异（或线性）

表3：各Simon32变体中能够构建差分/线性辨别器的最大轮数（“D/L”列）与能够成功实施NN攻击的最大轮数（“NN”列）的比较。

组	D/L	神经	旋转参数
$\mathcal{A}$	15	27	(13,8,13)
		26	(9,8,9),(8,7,8)
		25	(8,1,1)
		13	(4,1,4),(5,4,5),(10,7,10),(12,1,12),(12,3,3),(14,5,14),(15,4,4),(15,10,10),(15,12,12)
		12	(4,3,3),(5,2,2),(5,2,5),(9,2,2),(10,1,1),(10,9,10),(11,2,2),(13,10,10)
		11	(2,1,1),(2,1,2),(3,2,2),(3,2,3),(4,3,4),(5,2,2),(6,1,1),(6,3,6),(6,5,6),(7,2,2),(7,6,7)
$\mathcal{B}$	11	7	(2,1,8),(3,2,8),(4,1,8),(4,3,8),(5,2,8),(5,4,8),(6,1,8),(6,3,8),(6,5,8),(7,2,8),(7,4,8),(7,6,8),(9,2,8),(9,4,8),(9,6,8),(10,1,8),(10,3,8),(10,5,8),(10,7,8),(12,5,8),(12,11,8),(13,4,8),(14,1,2),(14,11,8),(14,13,8),(15,6,8),(15,10,8)
		4	(11,6,8),(11,10,8),(13,6,8),(14,7,8),(15,4,8)
$\mathcal{C}$	8	9	(12,5,4)
		8	(4,1,12),(5,4,12),(12,1,4)
		5	(2,1,10),(4,3,11),(4,3,12),(9,2,1),(9,8,1)
		4	(4,1,9)
		2	(3,2,4),(3,2,11),(3,2,13),(3,2,14),(4,1,15),(8,7,9),(9,2,12),(9,4,1),(9,6,1),(10,1,12),(10,7,2),(11,8,12),(11,10,3),(13,2,5),(14,1,15),(14,5,13),(15,8,7),(15,10,12)
		1	(2,1,9),(3,2,12),(6,1,4),(7,6,12)
$\mathcal{D}$	7	3	(3,2,5)
		2	(2,1,6),(2,1,11),(3,2,1),(3,2,7),(4,1,2),(4,1,11),(4,1,14),(5,4,7),(5,4,14),(6,1,5),(6,5,7),(7,2,6),(7,2,13),(7.4.13),(8,1,11),(8,5,15),(8,5,7),(9,2,3),(9,4,2),(9,4,3), (9,8,13),(10,1,5),(10,3,5),(10,9,5),(11,2,1),(11,2,9),(11,8,15),(15,12,9),(15,12,10),(15,14,10),(15,14,13)

从表3中，我们可以看出Simon32变体的NN攻击具有以下特征：

对于属于B组和D组的Simon32变体，NN攻击无法优于差分和线性攻击。

对于属于A组和C组的部分Simon32变体，神经网络攻击的表现可能优于差分攻击和线性攻击。具体而言，A组中具有（8,1,1）、（8,7,8）、（9,8,9）和（13,8,13）的Simon32变体，以及C组中的（12,5,4）均属于这种情况。

我们将在第3.3节中对这些特征进行更深入的探讨。

实验验证。在Simon32变体的实验中，我们通过平均匹配率来判断攻击是否成功。为验证该条件的正确性，我们对所有Simon16变体进行了与上述相同的神经网络攻击实验。我们可以用精确匹配率作为判断是否成功的条件

本研究关注的是差异性（或线性）特征的概率，而非其聚类效应。

表4：各Simon16变体中能够构建差分/线性辨别器的最大轮数（“D/L”列）与能够成功实施NN攻击的最大轮数（“NN”列）的比较。

组	D/L	神经	旋转参数
$\mathcal{A}$	15	29	(4,1,1),(5,4,4),(5,4,5),(7,0,0)
		28	(1,0,0),(3,0,0),(5,0,5),(7,0,7)
		27	(3,0,3),(4,1,4),(5,0,0)
		26	(1,0,1),(4,3,3),(4,3,4),(7,4,7)
		24	(7,4,4)
		15	(6,1,1),(7,6,7)
		14	(2,1,2),(6,5,6)
		13	(2,1,1),(3,2,2),(5,2,2),(5,2,5),(6,3,6),(6,5,5),(7,2,2),(7,2,7),(7,6,6)
		12	(3,2,3),(6,1,6),(6,3,3)
		7	(2,1,0)
$\mathcal{B}$	11	10	(4,3,0),(5,0,4)
		9	(1,0,4),(3,0,4),(3,2,0),(4,1,0),(5,2,4),(5,4,0),(6,5,4),(7,0,4),(7,4,0)
		8	(2,1,4),(3,2,4),(6,1,0),(6,5,0),(7,6,4)
		7	(5,2,0),(6,1,4),(6,3,0),(6,3,4),(7,2,0),(7,2,4),(7,6,0)
$\mathcal{C}$	8	10	(3,2,6)
		9	(6,3,2),(7,2,6)
		8	(2,1,6),(2,1,7),(5,2,6),(6,5,2),(7,6,1)
		7	(6,1,2)
		5	(3,0,7),(4,1,5),(5,2,3),(5,4,1),(6,1,7),(6,3,5),(7,0,3),(7,2,1)
		4	(1,0,3),(1,0,5),(1,0,7),(2,1,5),(3,0,1),(3,0,5),(3,2,5),(3,2,7),(4,1,7),(4,3,7),(5,0,1),(5,2,1),(5,4,3),(6,1,5),(6,3,7),(6,5,1),(6,5,3),(7,2,3),(7,4,3),(7,6,3)
		3	(3,0,2),(4,3,5),(5,0,3),(5,4,6),(7,0,1),(7,4,1),(7,6,1)
		2	(1,0,2),(1,0,6),(3,0,6),(4,1,2),(4,1,6),(4,3,2),(4,3,6),(5,0,2),(5,0,6),(5,4,2),(7,0,2),(7,0,6),(7,4,2),(7,4,6)
$\mathcal{D}$	7	4	(6,1,3),(6,3,1),(6,5,7),(7,0,5),(7,2,5),(7,4,5),(7,6,5),(2,1,3),(3,2,1),(4,1,3),(4,3,1),(5,0,7),(5,2,7),(5,4,7)

在实验中对Simon16变体进行了攻击；因此，如果Simon16变体的实验结果与Simon32变体的实验结果相似，则我们的Simon32变体分析可以视为有效。

我们的实验结果与Simon32变体的分类方式相同，被划分为四个组别，具体如表4所示。通过分析该表格可以发现，针对所有Simon16变体的神经网络攻击都具有以下特征：

对于属于B组和D组的Simon16变体，NN攻击无法优于差分和线性攻击。

对于属于A组和C组的Simon16变体的一部分，NN攻击可以优于差分和线性攻击。更具体地说，A组中具有（1,0,0）、（1,0,1）、（3,0,0）、（3,0,3）、（4,1,1）、（4,1,4）、（4,3,3）、（4,3,4）、（5,0,0）、（5,0,5）、（5,4,4）、（5,4,5）、（7,0,0）、（7,4,4）和（7,4,7）的Simon16变体，以及C组中具有（3,2,6）、（6,3,2）和（7,2,6）的变体，都属于这种情况。

综上，Simon16变体的实验结果与Simon32变体的实验结果相似，因此可以得出结论，Simon32变体的实验结果可视为有效。

3.3 讨论

在本小节中，基于第3.2节给出的实验结果，我们讨论Simon32变体的哪些结构容易受到NN攻击，以及NN攻击检测到哪些类型的漏洞。

识别Simon32变体的易受攻击结构。在第3.2节中，我们已明确指出：针对组A中的（8,1,1）、（8,7,8）、（9,8,9）和（13,8,13）以及组C中的（12,5,4）的Simon32变体，神经网络攻击的表现优于差分攻击和线性攻击。此外，我们对Simon16变体也获得了类似结论。这些结果表明，相较于差分攻击和线性攻击，Simon32变体（以及Simon16变体）之所以更容易遭受神经网络攻击，主要具备以下特征：

在A组旋转参数中，包含“a=c”或“b=c”的Simon32（及Simon16)变体比差分攻击和线性攻击更容易遭受神经网络攻击。例如，采用（a=c）参数的Simon32变体包括（8,7,8）、（9,8,9）和（13,8,13）三种组合，而采用（b=c）参数的则为（8,1,1）。Simon16变体也存在类似情况。

在C组旋转参数中，包含“n-a=c”或“n-b=c”的西蒙32（及西蒙16）变体比差分攻击和线性攻击更容易遭受神经网络攻击。例如，具有（12,5,4）参数的西蒙32变体对应“n-a=c”的情况；同理，（6,3,2）参数的西蒙16变体对应“n-a=c”；而（3,2,6）和（7,2,6）参数的西蒙16变体则对应“n-b=c”的情况。

可以看出，A组中具有这种旋转参数的Simon32变体最容易导致针对NN攻击的漏洞；然后，我们对这些Simon32变体进行全面分析，并阐明NN攻击检测到的漏洞类型。然而，关于C组的上述旋转参数也很有趣，因此，对这些参数的详细分析将是我们的未来工作。

对易损结构的综合分析。我们从以下两个方面对具有易损结构（即其旋转参数包含“a=c”或“b=c”）的Simon32变体进行综合分析：

——我们仅对旋转参数属于A组的32个Simon32变体进行了神经网络攻击实验，如第3.2节所示。随后，我们以与先前实验相同的方式，对旋转参数仍属A组的所有剩余Simon32变体进行了额外的神经网络攻击实验。

表5：各Simon32变体在满足“a=c”或“b=c”旋转参数条件下，其最大轮数对比表。其中“D/L”列表示可构建差分/线性（D/L）区分器的最大轮数，“ID/ZC”列表示无法构建差分/零相关线性（ID/ZC）区分器的最大轮数，“I”列表示可成功实施神经网络攻击（NN）的最大轮数。

组	D/L	身份证/ZC	I	神经	旋转参数
$\mathcal{A}$	15	17	32	29	(1,0,0)
				28	(1,0,1),(8,7,7),(11,8,11),(13,8,8),(15,0,0),(15,0,15)
				27	(3,0,0),(3,0,3),(5,0,0),(5,0,5),(7,0,0),(7,0,7),(8,1,8),(8,3,3),(8,3,8),(8,5,5),(8,5,8),(9,0,0),(9,0,9),(9,8,8),(11,0,11),(11,8,8),(13,0,0),(13,8,13),(15,8,8),(15,8,15)
				26	(8,7,8),(9,8,9),(11,0,0),(13,0,13)
				25	(8,1,1)
			20	13	(2,1,1),(2,1,2),(5,2,5),(6,3,3),(10,5,5),(13,2,13),(14,7,14),(15,14,14),(15,14,15)
				12	(5,2,2),(9,2,2),(9,2,9),(10,1,1),(10,1,10),(10,5,10),(10,9,9),(10,9,10),(11,6,11),(13,10,10),(13,10,13),(14,11,11),(14,11,14),(15,6,15)
				11	(6,3,6),(7,6,6),(7,6,7),(11,6,11),(13,2,2),(14,3,3),(14,3,14),(14,7,7),(15,6,6)
		13	20	13	(4,1,1),(4,1,4),(5,4,4),(5,4,5),(7,4,4),(7,4,7),(9,4,4),(9,4,9),(11,4,4),(11,4,11),(12,1,1),(12,1,12),(12,3,3),(12,3,12),(12,5,5),(12,5,12),(12,7,7),(12,7,12),(12,9,9),(12,9,12),(12,11,11),(12,11,12),(13,4,4),(13,4,13),(15,4,4),(15,4,15),(15,12,12),(15,12,15)
				12	(13,12,13)
				11	(4,3,3),(4,3,4),(13,12,12)
			18	13	(6,5,5),(9,6,6),(9,6,9),(10,3,3),(10,3,10),(10,7,7),(10,7,10),(11,2,11),(11,10,10),(11,10,11),(13,6,13),(14,5,14),(14,13,13),(14,13,14),(15,2,2),(15,10,10)
				12	(7,2,7),(11,2,2),(13,6,6),(14,1,14),(14,9,9),(14,9,14),(15,10,15)
				11	(3,2,2),(3,2,3),(6,1,6),(6,5,6),(7,2,2),(14,1,1),(14,5,5),(15,2,15)

-我们从差异攻击、线性攻击、不可能差异攻击、零相关线性攻击和积分攻击的角度比较了能力的差异。

表5展示了各Simon32变体中，能构建差分/线性（“D/L”列）、不可能差分/零相关线性（“ID/ZC”列）及积分型（“I”列）区分器的最大轮数，以及神经网络攻击（“NN”列）可成功实施的最大轮数对比。其中“组别”列定义如第3.1节所述。通过该表格我们可以观察到以下特征：

- NN攻击很少能检测到不可能的差分和零相关线性攻击造成的漏洞。
-神经网络攻击有可能检测到积分攻击造成的漏洞。具体来说，对于Simon32变体中构建积分区分器的最大轮数为32的情况，成功实施神经网络攻击所需的轮数相对较高，范围在25到29之间。

其中值得注意的一点是，这种情况下旋转参数都包含‘0’或‘8’（=n/2）。

要准确识别神经网络攻击为何会检测到这些特征仍具挑战性，这仍是未来研究的重点。简而言之，我们通过神经网络攻击发现，包含“a=c”或“b=c”以及‘0’或‘8（=n/2）’的Simon32变体结构最为脆弱。虽然神经网络攻击方法相对简单，但它们能从对称密钥密码的五种通用攻击角度（即差分攻击、线性攻击、不可能差分攻击、零相关线性攻击和积分攻击）识别出易受攻击的结构。这一发现表明，神经网络攻击可作为设计安全对称密钥密码（尤其是类Simon密码）的重要工具。

4 仔细观察脆弱的结构

在本节中，我们将仔细查看具有脆弱结构的Simon32变体，包含“a=c”或“b=c”以及‘0’或‘8’（=n/2）。

4.1 包含“a=c”或“b=c”的旋转参数的影响

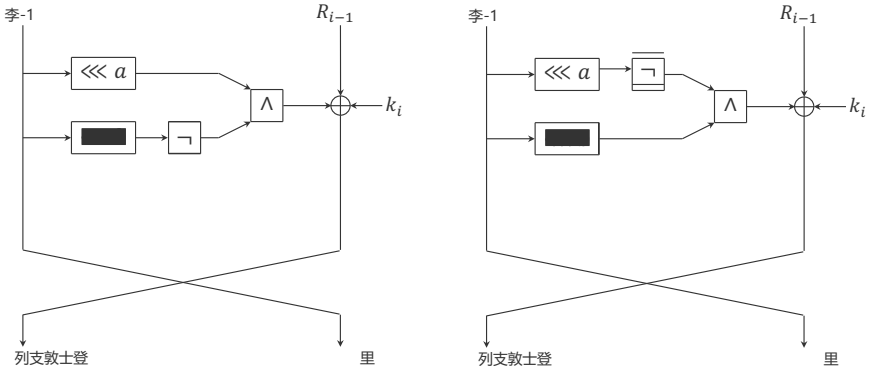
我们首先研究当Simon32变体中的旋转参数包含“a=c”或“b=c”时，f函数的结构会发生怎样的变化。例如，当Simon32变体的旋转参数包含“a=c”时，由公式等式(1)表达的f函数可以转换为

$$\begin{aligned}\hat{f}(x) &= \big((x \lll a) \wedge (x \lll b)\big) \oplus (x \lll a) \\ &= (x \lll a) \wedge \big((x \lll b) \oplus (11\ldots11)\big) \\ &= (x \lll a) \wedge \neg(x \lll b), \end{aligned} \tag{5}$$

其中¬表示逻辑非运算符。对于包含“a=c”的Simon32变体，其ˆf函数也可采用与公式(5)相同的转换方式。值得注意的是，当Simon32变体中的旋转参数包含“a=c”或“b=c”时，如图2所示，其ˆf函数只能通过AND操作来表达。基于此，我们在第4.2节进一步分析了不仅包含“a=c”或“b=c”，还包含‘0’或‘n/2’的旋转参数的影响，并在第4.3节后续小节中探讨了该函数的偏差输出特性。

4.2 包含60‘或6n/2’的旋转参数的影响

这里，我们重点讨论扩散特性，即明文比特对中间轮次内部状态产生影响的扩散程度。如果扩散特性较低，则在大轮次中，明文比特与密文比特之间会产生相关性。我们推测NN攻击可能会恢复



(a) 旋转参数：“a=c”。 (b) 旋转参数：“b=c”。

图2：当Simon32变体中的旋转参数包含“a=c”或“b=c”时的圆函数。

利用大轮次中的相关性，以比随机搜索更高的概率从密文比特中提取明文比特。

在本节中，我们验证了L_0的最低有效位（LSB）的影响在满足“a=c”或“b=c”的若干旋转参数下易于传播，其中包括‘0’或‘n/2’[10]。

本地观测。我们首先聚焦于（1,0,1）的旋转参数，该参数包含“a=c”和‘0’，重点分析L_0的最低有效位向两轮内部状态传播效应的易感性。图3展示了L_0的最低有效位如何扩散至（L_2，R_2）。在第一轮传播中，L_0的最低有效位影响了L_1的最后两位以及R_1的最低有效位。随后，L_0的最低有效位（LSB）所包含的影响会通过L_1的0¯位左移运算，传递至R_1的LSB。这意味着经过两轮运算后，明文比特的影响范围会缩小一个比特。接下来我们关注旋转参数（8,1,1），该参数包含“b=c”和‘n/2’。与前例类似，明文比特的影响范围在两轮运算后也会缩小一个比特。具体而言，L_0的LSB影响会通过L_0的（n/2）位左移运算，传递到R_0的第（n/2）低位（即第16位）。最后，L_1的第（n/2）低位中包含的L_0的LSB影响，会通过L_1的n/2位左移运算，最终融入R_1的LSB中。

长期圆周观测。最后，我们针对旋转参数（1,0,1）、（5,2,2）和（1,8,2）进行实验，以评估其易用性。

[10] 为简化说明，我们以L_0的最低有效位的影响为例，但显然，通过考虑R_0中某一位的影响，满足扩散特性的轮数可以增加一个。

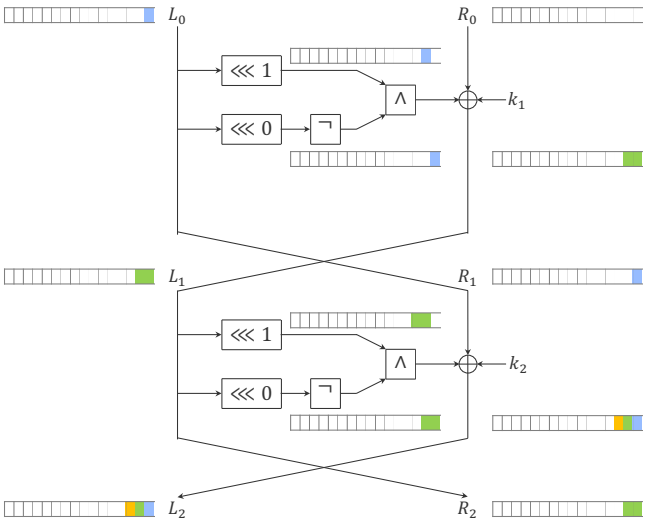


图3：（1,0,1）旋转参数的扩散特性。

关于将明文比特的影响传播到长期轮次的内部状态。需要注意的是，（5,2,2）的旋转参数包含“b=c”但不包含‘0’或‘n/2’，而（1,8,2）的旋转参数则适用于原始的Simon32算法。

在深入探讨实验结果细节之前，我们先说明计算机模拟的具体流程。该模拟通过计算每个轮次内部状态中各比特位受明文位影响的概率来进行研究。为此，我们制定了以下五条传播规则，用于概率性评估特定比特位的影响扩散情况：

规则1（初始化）。目标明文位被设置为1.0，其余位被设置为零。

规则2（轮转/交换）。概率，包括轮转和交换操作中的明文位的影响，根据目标操作进行传播。

规则3（AND）。在AND操作中，仅当一个输入位值为1时，whose probability is assumed to be $\frac{1}{2}$, the effect of the plaintext bits in the另一个输入位的对应输出位随之传播。随后，概率值（包含两个输入位在与操作中明文位的影响）将乘以$\frac{1}{2}$ and propagates to the correspond输入输出位。

规则4（异或运算）。在异或运算中，存在一种情况：明文位的影响可能不依赖于对应的两个输入位而直接传递到输出位。此时，包含异或运算两个输入位明文位影响的概率会被累加并传递到对应的输出位。若最终概率值超过1.0，则会被修正为1.0。

表6：（1,0,1）旋转参数的仿真结果。我们只将L_0的最低有效位设置为1.0，并检验其对r轮内部状态的影响。

比特位置	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L²	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
R_0	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
L_1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	
R_1	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
六	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.25	0.50	
R_2	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50
⋮							⋮									
L_{22}	0.65	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	
R_{22}	1.00	0.27	0.84	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
L_{23}	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	
R_{23}	0.65	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
L_{24}	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	
R_{24}	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

规则5（停止）。当所有概率，包括明文位的影响，都变为1.0时，模拟停止。

该评估旨在尽可能直观地展示明文位效应的传播情况，并重点关注影响输出的概率变化。因此，不考虑效应的抵消。

表6列出了我们对Simon32变体（1,0,1）的模拟结果。
表格显示我们的模拟在24轮后停止，因此目标变体被认为在24轮内满足扩散特性。我们还对旋转参数为（5,2,2）和（1,8,2）的实验进行了相同评估，结果分别列于表7和表8中。这些表格表明，旋转参数为（5,2,2）和（1,8,2）时，满足扩散特性的轮数分别为13轮和7轮。综上所述，我们的实验结果表明，当旋转参数包含‘0’或‘n/2’且满足‘a=c’或‘b=c’时，Simon32变体的扩散特性会减弱。

4.3 f函数输出值偏移的影响

本小节将阐明函数ˆf输出值存在偏差的影响。为此，我们主要研究具有“a=c”特性的Simon32变体情况，并首先考虑等式的以下位表示形式(5)：

$$\hat{f}_j(x) = (x \lll a)_j \wedge \neg(x \lll b)_j, \tag{6}$$

表7：（5,2,2）旋转参数的仿真结果。我们只将L_0的最低有效位设置为1.0，并检验其对r轮内部状态的影响。

比特位置	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L²	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
R_0	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
								0.00								
L_1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.50	0.00	
R_1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
								1.00								
六	0.00	0.00	0.00	0.00	0.00	0.25	0.00	0.00	0.50	0.00	0.00	0.25	0.00	0.00	0.00	
R_2	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.50	0.00
								0.00								
⋮								⋮								
L_{11}	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.79	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
R_{11}	1.00	1.00	0.12	1.00	1.00	1.00	1.00	1.00	1.00	0.93	1.00	1.00	0.51	1.00	1.00	1.00
L_{12}	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
R_{12}	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.79	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
L_{13}	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
R_{13}	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

表8：（1,8,2）旋转参数的仿真结果。我们只将L_0的最低有效位设置为1.0，并检验其对r轮内部状态的影响。

比特位置	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L_0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
十</sup>	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
																0.00
L_1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.00	1.00	0.50	
第$_1$页	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
																1.00
L_2	0.00	0.00	0.00	0.00	0.00	1.00	0.50	0.00	0.00	0.00	0.00	1.00	1.00	0.25	0.00	
十四	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.00	1.00	0.50
																0.00
L_3	0.00	0.00	0.00	1.00	1.00	0.38	0.00	1.00	0.00	1.00	1.00	0.75	0.13	1.00	1.00	
第$_3$页	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.50	0.00	0.00	0.00	0.00	1.00	1.00	0.25	0.00
																1.00
L_4	0.00	1.00	1.00	1.00	0.25	1.00	1.00	1.00	1.00	1.00	0.50	1.00	1.00	0.94	0.00	
18	1.00	0.00	0.00	0.00	1.00	1.00	0.38	0.00	1.00	0.00	1.00	1.00	0.75	0.13	1.00	1.00
																0.00
L_5	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.72	1.00	1.00	1.00
R_5	0.00	1.00	1.00	1.00	0.25	1.00	1.00	1.00	1.00	1.00	0.50	1.00	1.00	0.94	0.00	
																1.00
L_6	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
R_6	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.72	1.00	1.00	1.00

L_7	1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
R_7	1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00

其中$_j$表示x值的第j位，j。 {0,1 ，...，n−1}[11]。根据公式(6)中的条件，当且仅当满足以下条件时，函数ˆfj(x)=1成立：即存在集合$_j$ ю$_j$，使得$_j$ =（x。 a）$_j$ = 1且$_j$ =（x。 b）$_j$ = 0。

[11] 需要注意的是，L和R的索引代表回合数。

设p为输入x与函数ˆf的比值x_j = 1。假设在所有比特位置上xj = 1的概率（j 。{0,1}，...，n—1）是无偏的，我们可推导出以下方程：

$$\Pr\big((x \lll a)_j = 1\big) = \Pr\big((x \lll b)_j = 1\big) = p. \tag{7}$$

根据等式(7)，满足ˆfj(x)=1的概率如下：

$$\begin{aligned}\Pr\big((x \lll a)_j \wedge \neg(x \lll b)_j = 1\big) &= \Pr\big(((x \lll a)_j = 1) \wedge ((x \lll b)_j = 0)\big)\\ &= \Pr\big((x \lll a)_j = 1\big) \cdot \Pr\big((x \lll b)_j = 0\big)\\ &= p \cdot (1-p).\end{aligned} \tag{8}$$

由于0≤p≤1，显然由等式(8)表示的概率始终低于14。

换句话说，通过利用函数ˆf中的这个有偏事件，可以计算出当j。{0,1，...，n—1}时，函数ˆfj(x)=0的概率。is at least $\frac{3}{4}$基于这些考量，我们可以假设函数ˆfj（L_{i-1}）=0以较高的概率成立，此时第i轮函数可表示为

$$\text{Li} = \hat{f}\ (\text{Li}_{-1}) \quad \text{Ri}_{-1} \quad \text{ki} = \text{Ri}_{-1} \quad \text{ki}, \tag{9}$$

$$R_i = L_{i-1}. \tag{10}$$

寻找有效的线性近似方法。我们假设在2r轮西蒙32变体的易受攻击结构中，当i 。{2,4，...，2（r−1）}时，满足ˆfj（L_{i-1}）=0，即除了最后一轮外的所有偶数轮次均成立。在已知明文攻击场景下，攻击者可以获得多个已知明文/密文对。若上述假设成立，则攻击者可利用以下方程推导第j位的表达式：

$$(L_0 \oplus L_{2r} \oplus \hat{f}(R_{2r}))_j = \bigoplus_{x \in \{1,2,\ldots,r-1\}} (k_{2x})_j, \tag{11}$$

其中L_0表示明文的左侧，（L_2r，R_2r）表示密文。因此，等式（11）表示由明文比特、密文比特和子密钥比特线性和构成的线性近似表达式。为便于理解，我们提供了图示方式来可视化由等式（11）表示的四轮线性近似过程，如图4所示。在此，我们给出以下定理。

定理2.对于具有脆弱结构的2r轮Simon32变体，由公式等式（11）表示的线性近似成立的概率可表示为

$$\Pr(X = Y) = \begin{cases} 1 & \text{其中，ifo}\ t^{\text{rh}}e^{=\ 1\text{rwi}}s^{;} \\ \sum_{i=0}^{\lfloor \frac{r-1}{2} \rfloor} \binom{r-1}{2i} \left(\frac{1}{4}\right)^{2i} \left(\frac{3}{4}\right)^{r-2i-1} & \end{cases}$$

e'，(12)，其中X=(L_0 L_{2r} ˆf (R_{2r}) j，Y = 田 x, {1，2，⋯,r−1} (k_{2x}) j.

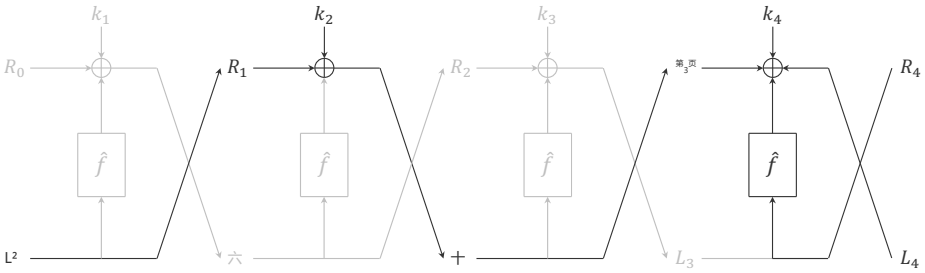


图4：以等式（11）表达的四轮线性近似示意图。我们假设在除最后一轮外的所有偶数轮中，函数ˆfj（L_{i-1}）=0成立。为此，我们的线性近似模型未考虑灰色线条。

证明。我们首先针对r = 1的情况进行推导。在这种情况下，我们可以推导出

$$\left(L_0 \oplus L_2 \oplus \hat{f}(R_2)\right)_j = (k_2)_j \tag{13}$$

来自等式（11），显然以概率1成立。

接下来，我们针对r = 2的情况进行分析。为了计算目标事件的概率，我们需要考虑第二轮中未知的输出值ˆf函数，即ˆfj（L_1）。若此时ˆfj（L_1）=0，则其成立的概率为

approximately $\frac{3}{4}$, the target event occurs with a probability of 1; otherwise, it 永远不会发生。然后，我们可以得到近似概率$\frac{3}{4}$ when $r = 2$.

最后，我们针对r > 2的情况进行研究，但该结论可推广至r = 2的情形。与r = 2的情况类似，我们需要在偶数轮次（不包括最后一轮）中考虑函数ˆf的未知输出值，即ˆfj（L_1）、ˆfj（L_3），...，ˆfj（L_{2r-3}）。换言之，需要考虑的函数数量为r-1。此时，目标事件以概率1发生的条件可表述为

$$\hat{f}_j(L_1) \oplus \hat{f}_j(L_3) \oplus \cdots \oplus \hat{f}_j(L_{2r-3}) = 0, \tag{14}$$

且该条件在以下两种情况下成立：

案例1.目标函数ˆf的输出值均为零。

案例2.目标函数ˆf的输出值为1的数值是偶数。

考虑到这两个情况，我们可以得到

$$\begin{aligned} &\binom{r-1}{0}\left(\frac{1}{4}\right)^0\left(\frac{3}{4}\right)^{r-1} + \binom{r-1}{2}\left(\frac{1}{4}\right)^2\left(\frac{3}{4}\right)^{r-2-1} + \cdots \\ &= \sum_{i=0}^{\lfloor \frac{r-1}{2} \rfloor} \binom{r-1}{2i}\left(\frac{1}{4}\right)^{2i}\left(\frac{3}{4}\right)^{r-2i-1} \end{aligned} \tag{15}$$

当r≥2时，证明结束。 □

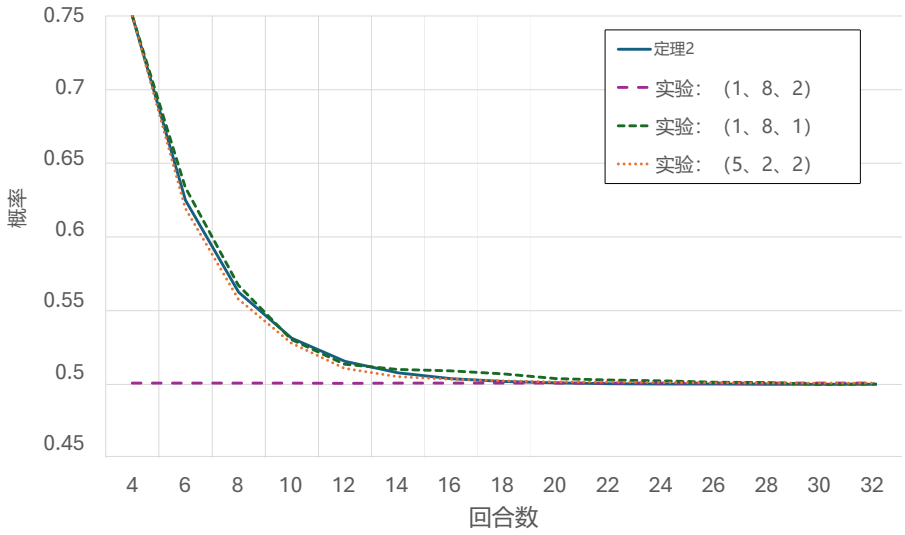


图5：理论概率与实验概率的比较。

定理2的实验验证。我们验证了定理2中理论值的准确性。为此，我们进行了2^8次试验（即随机生成2^8个密钥），每次试验使用2^{28}个样本。

图5展示了理论概率与实验概率的对比结果。纵轴表示概率值，横轴则对应目标密码的轮数。蓝色曲线代表定理2给出的理论概率，紫色、绿色和橙色点线分别对应旋转参数为（1,8,2）、（1,8,1）[12]和（5,2,2）的Simon32变体实验概率。

从图中可以看出，对于具有易损结构的Simon32变体（即旋转参数为（1,8,1）和（5,2,2））,理论值与实验值基本吻合。基于上述原因，可以认为定理2给出的理论值是正确的。此外，从图表中还可以看出，原始Simon32算法（即（1,8,2）的旋转参数）的理论值已趋近于随机概率。这一现象表明，原始Simon32算法与具有脆弱结构的Simon32变体在随机性方面存在显著差异。

构建线性判别器。由公式等式（11）表达的线性近似方法使攻击者能够构建有效的线性判别器。表9列出了线性近似理论概率及数据complexity__for__，其中每个判别器的成功概率为0.7。

[12] 参数（1,8,1）的旋转参数等于参数（8,1,1），这与原始参数（1,8,2）非常相似。

表9：用等式（11）表示的线性近似理论概率以及在每个偶数轮中成功概率为0.7的线性区分器构建数据复杂度。

周围	概率	数据（$_2$对数）	成功率
4	0.750000	5.164	0.7
6	0.625000	7.164	0.7
8	0.562500	9.164	0.7
10	0.531250	11.164	0.7
12	0.515625	13.164	0.7
14	0.507812	15.164	0.7
16	0.503906	17.164	0.7
18	0.501953	19.164	0.7
20	0.500976	21.164	0.7
22	0.500488	23.164	0.7
24	0.500244	25.164	0.7
26	0.500122	27.164	0.7
28	0.500061	29.164	0.7
30	0.500030	31.164	0.7
32	0.500015	33.164	0.7

这些基于定理1和2推导出的偶数轮线性判别器，从表格中可以看出，针对存在漏洞结构的Simon32变体，最多可构建30轮的线性判别器。鉴于表5所示神经网络攻击在29轮内已成功实施，这表明此类神经网络攻击可能检测到这类线性判别器。详细分析将是我们的未来研究重点。

其他线性近似方法。我们简要介绍三种线性近似方法。
这些运动与由等式（11）表达的线性近似非常相似。第一个线性近似由以下公式给出：

$$(R_0 \oplus \hat{f}(L_0) \oplus R_{2r})_j = \bigoplus_{x \in \{1,2,\ldots,r\}} (k_{2x-1})_j. \tag{16}$$

这可用于计算所有奇数轮次子密钥位（即k_1 k_3 … k_{2r-1}）的线性和，以用于偶数轮次（即。，2r轮次）的目标Simon32变体。

下一个线性近似由下式给出

$$(L_0 \oplus R_{2r+1})_j = \bigoplus_{x \in \{1,2,\ldots,r\}} (k_{2x})_j. \tag{17}$$

这对于计算所有偶数轮次的子密钥位的线性和是有用的（即，对于奇数轮次（即，2r+1轮次）的目标Simon32变体的k_2 k_4 … k_{2r}）。

最后一个线性近似由下式给出

$$(R_0 \oplus \hat{f}(L_0) \oplus L_{2r+1})_j = \bigoplus_{x \in \{1,2,\ldots,r,r+1\}} (k_{2x-1})_j. \tag{18}$$

这对于计算目标Simon32变体的奇数轮（即，2r +1轮）的所有奇数轮次的子密钥位的线性和是有用的（即，k_1 k_3 … k_{2r+1}）。

每个线性近似成立的概率也可以用类似于定理2的方法推导，但这超出了本文的范围。

5 针对具有脆弱结构的Simon32变体的密钥恢复

本节中我们主要使用由等式（11）表示的线性近似，但预期由方程（16）-（18）表示的其他线性近似也可用于我们的分析。

聚焦于等式（11），其表达式由明文位、密文位和子密钥位的线性和构成。这意味着我们找到了有效的线性近似方法，这正是线性密码分析[24]的主要目的。换句话说，这种线性近似可以用于开发基于线性密码分析的密钥恢复攻击。

线性密码分析最早由松井在1993年EUROCRYPT会议上提出[24]。他提出了两个简洁而强大的算法，即著名的松井算法1和算法2。其后续研究[25]改进了线性密码分析方法，并成功应用于破解完整16轮DES的首个计算机实验。此后，基于线性密码分析的密钥恢复攻击通过引入多种技术手段不断升级，例如沃尔什变换（或快速傅里叶变换）[11]、仿射剪枝沃尔什变换[12]、沃尔什频谱穿孔技术[13]等。

本文采用简单而强大的Matsui算法1，证明了针对存在漏洞结构的Simon32变体，其子密钥位的线性组合可被以较高概率成功恢复。虽然运用其他技术[11-13,25]可能带来进一步改进，但这些内容已超出本文研究范围。

5.1 重新审视Matsui算法1

本小节将简要回顾Matsui的算法1 [24]。线性密码分析的基本思想是为给定的密码寻找以下有效的线性近似：

$$P_{i_1} \oplus \cdots \oplus P_{i_a} \oplus C_{j_1} \oplus \cdots \oplus C_{j_b} = K_{k_1} \oplus \cdots \oplus K_{k_c}, \tag{19}$$

其中，Pi、Cj和Kk分别表示第i个明文比特、第j个密文比特以及第k个密钥（或子密钥）比特。同时，i_1，...，ia、j_1，...，jb、k_1

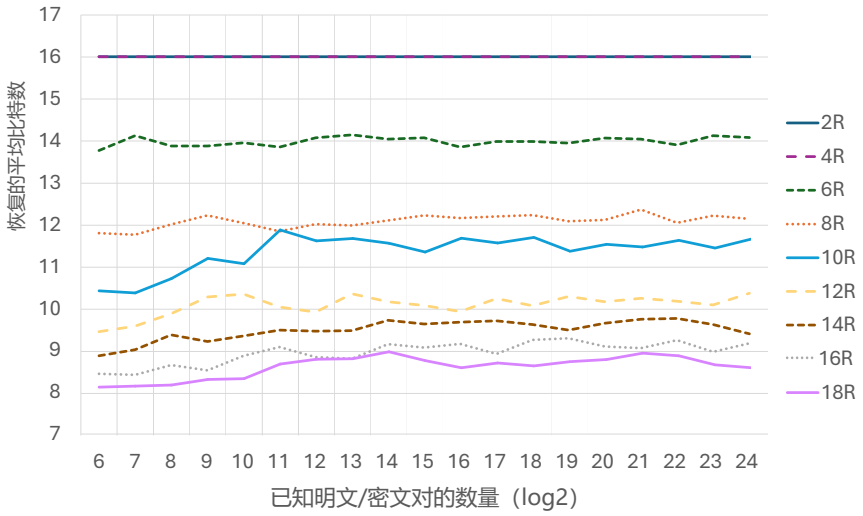


图6：对旋转参数为（5,2,2）的Simon32变体进行密钥恢复攻击的实验结果。

⋯ 和kc表示固定比特位置，等式（19）以概率p成立。$\neq \frac{1}{2}$)是随机生成的明文和相应的密文。

我们可以使用以下算法恢复子密钥位的线性和（即，ˆK=Kk$_1$ ⋯ Kk$_c$）：

步骤1.设T为等式（19）左侧等于零的明文数量，N为已知明文攻击场景下获得的明文/密文对的数量。

步骤2.如果T>$\frac{N}{2}$, then we guess 当p>时，K=0$\frac{1}{2}$ or 当p<时，K=1$\frac{1}{2}$否则，当p>时，我们推测K=1。$\frac{1}{2}$ or 当p<时，K=0$\frac{1}{2}$.

我们根据定理1来估计明文/密文对的数量以及基于Matsui算法1的攻击的成功概率。

5.2 实验验证

我们验证了基于松井算法1的关键恢复攻击对具有漏洞结构的Simon32变体的有效性。为此，我们进行了2^8次试验（即2^8个密钥）的实验，每次试验使用2^d个样本，其中d。{6,7，...，24}。以下针对r。{1,2，...，9}的两个2r轮次变体被纳入分析：第一个变体的旋转参数为（5,2,2），包含“b=c”但不包含‘0’或‘n/2’；第二个变体的旋转参数为（8,1,1），同时包含“b=c”和‘n/2’。

我们在图6和图7中提供了实验结果的详细信息。这些图表展示了对Simon32密钥恢复攻击的实验结果。

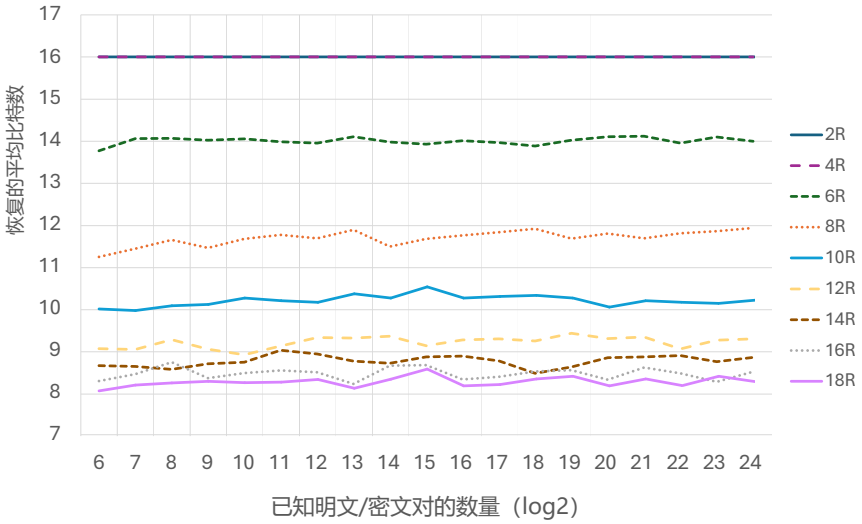


图7：对旋转参数为（8,1,1）的Simon32变体进行密钥恢复攻击的实验结果。

分别对应旋转参数为（5,2,2）和（8,1,1）的变体。具体而言，这些图表展示了不同已知明文/密文对数量下，16位子密钥线性叠加运算中恢复的平均比特数。从这些图表中可以观察到以下特征：

-对于最多四轮的变体，我们以1的概率恢复子密钥线性和的全部16位。

-对于超过六轮的目标变体，恢复的比特数平均值逐渐下降，达到与随机猜测相同攻击能力（即16位中有8位可以被恢复）的轮数几乎为18。

令人惊讶的是，增加已知明文/密文对的数量仅略微提高了攻击能力。我们推测，这一因素是由于Simon32变体的扩散特性，以及第4节讨论的易受攻击结构所致。2，但详细的分析是我们未来的挑战。

此外，我们还针对原始Simon32算法进行了相同实验验证，发现其密钥恢复攻击在最多两轮内仍具有效性——这属于基础级攻击。总体而言，针对具有易受攻击结构的Simon32变体所实施的此类攻击，相较于原始Simon32算法仍具有相对有效性。通过研究我们获得新认知：随着已知明文/密文对数量的增加，攻击能力并不会随之提升。

6 结论和未来挑战

我们提出了一项创新性进展，旨在将基于神经网络的输出预测（NN）攻击转化为设计安全对称密钥密码（尤其是西蒙式密码）的有效工具。通过分析发现，利用神经网络攻击评估目标密码的安全性具有重要价值。值得注意的是，该方法能精准识别目标密码的薄弱结构。这一优势在于，具备此类脆弱结构的类似密码可被直接排除在所有设计方案之外。

神经网络攻击（NN攻击）是检测系统漏洞的有效工具，但遗憾的是，单靠这种工具无法确定导致漏洞的攻击向量。因此，我们建议将神经网络攻击与自动搜索方法有机结合，充分发挥两者的优势。这样在设计安全对称密钥密码时，就能实现更可靠、更高效的分析流程。

我们希望通过应对以下五个挑战，使神经网络攻击成为设计安全对称密钥密码的重要工具：1）我们主要针对“a=a”或“b=b”的情况进行深入分析，以揭示潜在的漏洞结构。类似地，研究“n-a=a”或“n-b=a”的情形，或将为神经网络攻击的应用提供新视角。2）有趣的是，神经网络攻击可能通过积分攻击发现漏洞。在实验中，我们将训练数据量限制为2^{15}，从积分攻击方法的角度看，这一观察结果看似可疑。但若能证明其正确性，神经网络攻击的应用范围将进一步拓展。3）神经网络攻击成功的最大轮数与基于线性近似表达式等式（11）构建线性判别器所需轮数高度相似，这暗示神经网络攻击可能通过线性攻击发现漏洞。4）令人意外的是，增加已知明文/密文对的数量仅略微提升了密钥恢复攻击的能力。确定其根本原因可能有助于开发线性密钥恢复攻击。5）本研究仅针对AND-旋转-XOR密码进行分析。为实现主要目标，我们需要评估神经网络攻击对更多类型对称密钥密码的有效性，例如基于加法-旋转-XOR和算术化方向的密码。
密码

参考文献

1. 贝涅尔、T.、朱诺德、P.、沃登奈、S.：《我们能超越线性密码分析到什么程度？》收录于《ASIACRYPT计算机科学讲义》第3329卷，第432-450页，施普林格出版社（2004年）
2. 包志、郭军、刘敏、马磊、涂毅：《增强差分神经密码分析》。载于《ASIACRYPT (1)：计算机科学讲义》，第13791卷，第318-347页，施普林格出版社，2022年。
3. 包志、陆军、姚宇、张力：深度学习辅助密码分析研究新进展。载于《ASIACRYPT（第3届）》计算机科学讲义，第144卷，第436–467页，施普林格出版社，2023年。

4. 比奥利厄，R.、肖尔斯，D.、史密斯，J.、特里特曼-克拉克，S.、威克斯，B.、温格斯，L.：西蒙与斯佩克轻量级分组密码家族。《IACR密码学电子预印本档案》第404页（2013年）
5. 贝利尼（E.）、热罗（D.）、汉比策（A.）和罗西（M.）：一种密码无关的神经训练流程，可自动发现优质输入差异。《IACR对称密码学汇刊》2023年第3期，第184–212页（2023年）
6. 本纳米拉，A.、热罗，D.、佩兰，T.、谭，Q.Q.：基于机器学习的密码分析深度研究。收录于《欧洲密码学会议（第1届）》。载于计算机科学讲义系列第12696卷，第805-835页。施普林格出版社（2021年）。
7. 比汉姆（E.）、比留科夫（A.）与沙米尔（A.）：基于不可能差分的跳棋游戏31轮密码分析。收录于《欧洲密码学会议论文集·计算机科学讲义》第1592卷，第12-23页，施普林格出版社，1999年。
8. 比汉姆，E.，沙米尔，A.：des类密码系统的差分密码分析。收录于《CRYPTO：计算机科学讲义》第537卷，第2–21页，施普林格出版社（1990年）
9. 博格达诺夫，A.，里伊门，V.：零相关线性外壳与分组密码的线性密码分析。密码学描述与密码学杂志70(3)，369–383（2014）
10. 布尔拉，C.，大卫，N.，博伊西耶，R.H.，纳亚-普拉斯森西亚，M.《稳定胜快速：SPEEDY-7-192的全面突破》收录于《EUROCRYPT（第4卷）》。该文发表于《计算机科学讲义》第14007卷，第36-66页，由施普林格出版社于2023年出版。
11. 科拉德（B.）、斯坦达特（F.）、奎斯卡特（J.）：《改进松井线性密码分析的时间复杂度》。收录于《ICISC计算机科学讲义》第4817卷，第77-88页，施普林格出版社，2007年。
12. 弗洛雷斯-古铁雷斯，A.：基于仿射沃尔什变换剪枝的线性密钥恢复攻击优化。载于《ASIACRYPT（第4届）》计算机科学讲义系列第13794卷，第447–476页，施普林格出版社，2022年。
13. 弗洛雷斯-古铁雷斯，A.，特多，Y.：基于沃尔什谱穿孔的线性密钥恢复攻击改进方案。《IACR密码学电子预印本汇编》第151页（2024年）
14. 戈尔，A.：基于深度学习的轮缩减speck32/64攻击改进。载于《密码学》第2卷。计算机科学讲义，第11693卷，第150-179页。施普林格出版社（2019年）
15. 哈迪普尔，H.，艾希尔斯德，M.：基于单项式预测的WARP整体密码分析。《国际密码学研究会对称密码学汇刊》2022年第2期，第92–112页（2022年）
16. 哈迪普尔，H.，格哈尔特，S.，萨德吉，S.，艾希尔斯德，M.改进了对积分攻击、不可能差分攻击和零相关攻击的搜索方法，并将其应用于ascon、forkskinny、skinny、mantis、PRESENT和qarmav2等密码学方案。该研究成果发表于《IACR对称密码学汇刊》2024年第1期，页码234-325（2024年）。
17. 木村H.、江村K.、伊索贝T.、伊藤R.、小川K.、大石T.：基于深度学习的分组密码输出预测攻击。收录于《ACNS研讨会论文集·计算机科学讲义》第13285卷，第248-276页，施普林格出版社（2022年）。
18. Kimura，H.，Emura，K.，Isobe，T.，Ito，R.，Ogawa，K.，Ohigashi，T.：基于深度学习的弱SPN分组密码输出预测攻击研究——深入分析。
J. 信息过程，第31卷，第550-561页（2023年）
19. 克努森，L.：《Deal-a 128位分组密码》，发表于《复杂性》期刊第258卷第2期，第216页（1998年）
20. 克努森，L.R.与瓦格纳，D.A.合著的《积分密码分析》收录于《FSE计算机科学讲义

集》第2365卷，第112-127页，施普林格出版社，2002年。
21. 科尔布（S.）、莱安德（G.）、蒂森（T.）：《西蒙分组密码系列研究观察》。收录于《密码学》第1卷，计算机科学讲义丛书第9215卷，第161-185页，施普林格出版社，2015年。
22. 近藤K.、佐佐木Y.、藤田Y.、岩田T.：关于SIMON分组密码的设计原理：针对SIMON的积分攻击与不可能差分攻击

《IEICE电子通信与计算机科学基金会汇刊》第101-A卷第1期（2018年）第88-98页

23. 刘F、阿南德R、王L、迈尔W、伊索贝T：系数分组：破解查格里密码及其他。收录于《欧洲密码学会议（第4届）》论文集，计算机科学讲义系列第14007卷，第287-317页，施普林格出版社，2023年。
24. 松井，M.：DES密码的线性密码分析方法。收录于《欧洲密码学会议论文集·计算机科学讲义》第765卷，第386-397页，施普林格出版社（1993年）
25. 松井，M.：数据加密标准的首次实验密码分析。载于《CRYPTO》计算机科学讲义第839卷，第1-11页，施普林格出版社（1994年）
26. 穆哈，N.，普雷内尔，B.：探索arx的最优差分特性——在salsa20中的应用。密码学电子预印本档案，论文2013/328（2013），https://eprint.iacr.org/2013/328，https://eprint.iacr.org/2013/328
27. 穆哈（N.）、王（Q.）、顾（D.）与普雷尼尔（B.）：基于混合整数线性规划的差分密码分析与线性密码分析。收录于《密码学会议论文集·计算机科学讲义》第7537卷，第57–76页，施普林格出版社，2011年。
28. 坂本K、伊藤R、伊索贝T。并行SAT框架在差异特征聚类中的应用研究。载于《计算机科学讲义》第14201卷，第409-428页，施普林格出版社（2023年）。
29. 孙立、王伟、王敏：基于SAT方法加速差分与线性特征搜索。《国际密码学对称性会议汇刊》2021年第1期，第269-315页（2021年）
30. 王S.、冯D.、胡B.、关J.、石T.：针对全轮次FRIET的实用攻击。《IACR对称密码学汇刊》2022年第4期，第105-119页（2022年）