低延迟S盒的构造及双向低延迟性质研究

吴瑞宸[1,2],张 蕾[1,3*]
(1. 中国科学院软件研究所,北京 100190;2. 中国科学院大学,北京 100049;3. 密码科学技术全国重点实验室,北京 100878)

摘 要: 低延迟分组密码的设计是目前密码学研究中的热点之一,其中低延迟S盒的构造是设计中的重要研究方向. 本文基于低延迟门电路和两层树型结构,搜索不同延迟水平下具有一定密码学性质的低延迟平衡布尔函数及其拓展比特置换等价类;基于将低延迟布尔函数作为分量布尔函数构造向量布尔函数的方法,本文构造得到了不同延迟水平下的低延迟S盒,并给出延迟性质和硬件实现面积具有优势的S盒实例;此外,本文对低延迟的S盒集合与逆S盒集合匹配搜索具有双向低延迟性质的S盒,给出搜索得到的实例. 与PRINCE、MANTIS等其他低延迟分组密码中使用的4 bit S盒相比,本文构造的低延迟S盒在延迟水平上相较MANTIS降低了20%,与PRINCE相比降低了33%,在硬件实现面积上相较MANTIS减少了6.68%,与PRINCE相比减少了17.69%.

关键词: 低延迟分组密码;低延迟S盒;门电路;向量布尔函数;双向低延迟性质
基金项目: 中国科学院稳定支持基础研究领域青年团队计划(No.YSBR-035)
中图分类号: TP302 **文献标识码:** A **文章编号:** 0372-2112(2024)11-3769-11
电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20231047

Research on Construction of Low-Latency S-Boxes and Bidirectional Low-Latency Properties

WU Rui-chen[1,2], ZHANG Lei[1,3*]
(1. *Institute of Software Chinese Academy of Sciences*, *Beijing* 100190, *China*;
2. *University of Chinese Academy of Sciences*, *Beijing* 100049, *China*;
3. *State Key Laboratory of Cryptology*, *Beijing* 100878, *China*)

Abstract: The quest for low-latency block ciphers is a burgeoning area of interest within the cryptographic community, with the development of low-latency S-boxes standing as a pivotal avenue of exploration.Leveraging gate circuits of minimal latency and a novel two-layer tree structure, our study delves into the construction of balanced Boolean functions and their extended bit permutation equivalence classes that manifest desirable cryptographic properties across varied latency thresholds. Utilizing these low-latency Boolean functions as coordinate functions, we craft vectorial Boolean functions to construct S-boxes with low-latency.Our research not only furnishes S-boxes optimized for latency performance and hardware implementation area but also pioneers the amalgamation of low-latency S-box sets with their corresponding inverse sets, searching for S-boxes with bidirectional low-latency property.The low-latency S-box in our investigation outperform existing benchmarks and offer more choices, showcasing a latency reduction of 20% and 33% over MANTIS and PRINCE, alongside achieving a hardware area reduction of 6.68% compared to MANTIS and a substantial improvement of 17.69% against PRINCE.

Key words: low-latency block cipher; low-latency S-box; gate circuit; vectorial Boolean functions; bidirectional low-latency properties
Foundation Item(s): CAS Project for Young Scientists in Basic Research (No.YSBR-035)

1 引言

随着云存储、大数据等技术的快速发展,人们享受到数据共享带来的巨大便利,同时隐私数据的保护也成为了重要的挑战. 隐私计算技术可以很好地解决数据共享中的隐私问题,其中可信执行环境则是目前主流的隐私计算技术之一. 可信执行环境中处理器会对

收稿日期:2023-11-10;修回日期:2024-06-06;责任编辑:梅志强
*通讯作者:张蕾

内存中的数据进行加密,在处理器中构造一个安全的执行环境以保证其内部数据的安全性,常用的可信执行环境方案包含 Intel 的 Software Guard Extensions 技术[1]、AMD 的 Secure Encrypted Virtualization 技术[2]等.由于可信执行环境的安全性和性能依赖于底层密码算法安全性和硬件实现,硬件实现上具有低延迟性质的密码算法能够在保证数据安全的同时显著提升架构中软件运行的速度.因此密码设计者们对密码算法的延迟性能提出了新的需求,低延迟分组密码算法凭借其硬件实现上的优势也受到广泛关注.

针对低延迟性质的研究最早开始于2012年,加密硬件与嵌入式系统会议上 Knežević 等人[3]对常用分组密码在硬件实现中的低延迟行为进行了研究,并给出了低延迟密码设计中的一系列建议,其中就包括在相同条件下具有密码学强度的3 bit或者4 bit S盒在低延迟性能上会优于有更高比特输入输出的S盒;同年,第一个在硬件实现上针对延迟进行优化的分组密码算法 PRINCE 在亚密会上由 Borghoff 等人[4]提出,并在之后更新的 PRINCEv2[5]中通过对密钥调度和轮函数进行调整,提升安全级别的同时降低了 PRINCEv2 在面积、延迟和能耗方面的开销;2015 年 Banik 等人[6]提出了低能耗的分组密码算法 Midori,算法针对硬件实现的能耗对电路进行了优化,同时也表现出了低延迟的性质;随后在 2016 年 Beierle 等人[7]提出了 SKINNY 分组密码算法和它的低延迟版本 MANTIS,相比 PRINCE 在延迟和安全性方面都得到了改进;2017 年,受到 PRINCE 和 MANTIS 启发,针对内存加密和指针认证设计的轻量级可调整分组密码算法 QARMA[8]被提出,算法被广泛应用于高通公司的 ARMv8.3 产品的指针认证过程中,通过在使用前验证指针标签,可以实现控制流完整性和硬件辅助软件的安全防护;2021 年 Leander 等人[9]提出了名为 SPEEDY 的超低延迟分组密码算法,算法面向高端处理器中的内存加密环节,其硬件实现的加密算法的延迟极低;同年,Banik 等人[10]提出了低延迟的伪随机函数 Orthros,算法以减少全展开电路的延迟为目标设计,实现了当时低延迟密码原语中最优的延迟性质.

S盒作为分组密码算法的核心组件,在许多密码算法中是唯一提供非线性性质的部件,对算法的整体安全性和延迟有重要的影响.最初对S盒的研究聚焦于其密码学相关的性质,比如抵抗差分、线性攻击的能力.为了更好地归纳S盒的密码学性质,设计者们提出等价类的概念来对S盒进行分类,并给出了一系列的算法与S盒分类结果.2003 年 Biryukov 等人[11]提出解决线性等价与仿射等价问题的算法;Leander 等人[12]在2007 年对最优 4 bit S盒进行仿射等价的分类;在 FSE(Fast Software Encryption)2015 上 Zhang 等人[13]利用等价类划分给出了几类拥有最优差分、线性分析抵抗能力的 Platinum 分类.实际上,目前为止所有的6 bit以下的二次置换都已经根据仿射等价完全分类,在文献[14]中首次提出了对5 bit二次置换的分类.基于该工作,文献[15]提出了对 n bit到 m bit布尔函数的高效等价类划分算法,并实现了6 bit二次置换的完全分类以及 $n\leqslant 6$ 的所有平衡二次函数的分类.

传统的S盒构造方法更多地追求安全性方面的需求,通常先根据密码学性质构造具有优秀线性、差分性质的S盒,再使用启发式或是建模搜索的方法来寻找S盒的低延迟实现.元胞自动机被广泛应用于S盒的构造中,2022 年赵耿等人[16]基于元胞自动机构造了扰动时空混沌系统以提高生成动态S盒的安全性,次年柴进晋等人[17]同样通过元胞自动机设计了能够抵抗错误注入攻击的S盒.2016 年的 FSE 上 Ko[18]提出了基于 SAT 建模和求解器来搜索指定S盒最优硬件实现的方法,给出其最小门电路数、最低门电路深度等情况下的硬件实现.然而使用这类算法虽然能够得到S盒的最优硬件实现,却不能保证对所有的S盒都行之有效,因此算法设计者尝试先从门电路级来构建具有良好延迟性质的S盒,再筛选得到S盒的密码学性质.SPEEDY 算法[9]的核心为超低延迟的6 bit S盒,算法设计中作者使用低延迟与非门来构造分量布尔函数,其中分量布尔函数都使用两层树型门电路结构构造得到,由此得到的S盒在硬件实现的正向延迟极低,保证在加密过程中算法延迟极低,但是解密过程中的延迟表现要远差于加密过程,因此如何构造双向低延迟的S盒仍是需要解决的困难问题.基于 SPEEDY 中构造S盒的方法,2023 年的 FSE 上 Rasoolzadeh[19]提出了一种高效构造双射向量布尔函数的方法,并研究了使用这种方法构造的低延迟S盒的差分与线性性质.

本文的主要工作包括:(1)拓展使用低延迟门电路和两层树型结构构造超低延迟布尔函数的方法,使用更多类型的低延迟门电路,搜索得到不同延迟水平下,满足一定密码学性质的布尔函数及其等价类;(2)基于(1)得到的低延迟布尔函数及其等价类,构造不同延迟水平下满足密码学指标的向量布尔函数及其等价类,并给出硬件实现中有着优秀性质的单向低延迟S盒实例;(3)进一步匹配搜索双向低延迟S盒及其等价类,并给出其中延迟性质最优的双向低延迟S盒实例.

2 预备知识

这一节中,本文将根据文献[9]和文献[19]中的定义介绍与布尔函数、向量布尔函数、布尔函数等价类和布尔函数实现相关的基础概念,用以帮助我们更好地

理解后面章节使用的概念与方法.

2.1 布尔函数

定义1 布尔函数:从向量空间$\mathbb{F}_2^n$映射到二元域$\mathbb{F}_2$的函数.

定义2 真值表:令f为一个n bit的布尔函数,f的真值表为二元向量$\boldsymbol{T}_f\in\mathbb{F}_2^{2^n}$,对任意的$\boldsymbol{x}\in\mathbb{F}_2^n$,都有$\boldsymbol{T}_f[\boldsymbol{x}]=f(\boldsymbol{x})$.

布尔函数真值表的汉明重量也被称为布尔函数的重量,如果一个布尔函数的重量等于2^{n-1},即对于一半的$\boldsymbol{x}\in\mathbb{F}_2^n$,$f(\boldsymbol{x})=1$,这类布尔函数被称为平衡布尔函数.

定义3 代数正规型:令f为一个n bit的布尔函数,f的代数正规型表示为如下形式

$$f(\boldsymbol{x})=\oplus_{\boldsymbol{I}\in\mathbb{F}_2^n}a_{\boldsymbol{I}}\boldsymbol{x}^{\boldsymbol{I}}=\oplus_{\boldsymbol{I}\in\mathbb{F}_2^n}a_{\boldsymbol{I}}\left(\prod_{i=0}^{n-1}x_i^{I[i]}\right)$$

其中,x_i表示输入变量$\boldsymbol{x}$的第i个分量,$\boldsymbol{x}^{\boldsymbol{I}}$表示单项式$x_0^{I[0]}x_1^{I[1]}\cdots x_{n-1}^{I[n-1]}$.

定义4 代数次数:令f为一个n bit的布尔函数,其代数次数$\deg(f)$为代数正规型表达式中单项式的最大汉明重量,即

$$\deg(f)=\max_{I\in\mathbb{F}_2^n,a_I=1}\mathrm{hw}(I)$$

当布尔函数代数次数为1,2和3时,布尔函数分别为线性函数,二次函数和三次函数.

2.2 向量布尔函数

定义5 向量布尔函数:从n bit向量映射到m bit向量的函数.

在密码学中,向量布尔函数也被称为S盒,用于提供对称密码算法中需要的混淆部分功能,在分组密码中扮演着重要的角色. 令$\boldsymbol{F}=(f_0(x),f_1(x),\cdots,f_{m-1}(x)$为$n$ bit到m bit的向量布尔函数,$f_0(x),f_1(x),\cdots,f_{m-1}(x)$被称为$\boldsymbol{F}$的分量布尔函数. 对$\boldsymbol{\alpha}\in\mathbb{F}_2^n\backslash\{0\}$,布尔函数$\boldsymbol{x}\mapsto\langle\boldsymbol{\alpha},\boldsymbol{F}(\boldsymbol{x})\rangle$被称为$\boldsymbol{F}$的组件函数.

平衡性在向量布尔函数中的定义与布尔函数类似. 对于一个n bit到m bit的向量布尔函数$\boldsymbol{F}$,如果取遍$\mathbb{F}_2^n$中所有输入对应$\mathbb{F}_2^m$个输出每个都出现2^{n-m}次,则称该向量布尔函数为平衡的.

向量布尔函数的代数次数为所有分量布尔函数代数次数中的最大值. 根据向量布尔函数的代数次数,我们同样可以定义其中的线性函数、二次函数和三次函数.

线性度和差分均匀度是用于评估一个向量布尔函数抵御线性攻击和差分攻击最重要的两个密码学性质,其定义如下.

定义6 线性度:令$\boldsymbol{F}:\mathbb{F}_2^n\to\mathbb{F}_2^m$为$n$ bit到m bit的向量布尔函数,其线性度定义如下

$$\mathrm{lin}(\boldsymbol{F})=\max_{\boldsymbol{\alpha}\in\mathbb{F}_2^n,\boldsymbol{\beta}\in\mathbb{F}_2^m\backslash\{0\}}\left|\begin{matrix}\#\{\boldsymbol{x}\in\mathbb{F}_2^n|\langle\boldsymbol{\alpha},\boldsymbol{x}\rangle=\langle\boldsymbol{\beta},\boldsymbol{F}(\boldsymbol{x})\rangle\}\\-\#\{\boldsymbol{x}\in\mathbb{F}_2^n|\langle\boldsymbol{\alpha},\boldsymbol{x}\rangle\neq\langle\boldsymbol{\beta},\boldsymbol{F}(\boldsymbol{x})\rangle\}\end{matrix}\right|$$

定义7 差分均匀度: 令$\boldsymbol{F}:\mathbb{F}_2^n\to\mathbb{F}_2^n$为$n$ bit到m bit的向量布尔函数,其差分均匀度定义如下

$$\mathrm{uni}(\boldsymbol{F})=\max_{\boldsymbol{\alpha}\in\mathbb{F}_2^n,\boldsymbol{\beta}\in\mathbb{F}_2^n\backslash\{0\}}\#\{\boldsymbol{x}\in\mathbb{F}_2^n|\boldsymbol{F}(\boldsymbol{x})\oplus\boldsymbol{F}(\boldsymbol{x}\oplus\boldsymbol{\alpha})=\boldsymbol{\beta}\}$$

定义8 完全性:令$\boldsymbol{F}:\mathbb{F}_2^n\to\mathbb{F}_2^m$为$n$ bit到m bit的向量布尔函数$\boldsymbol{F}(x_0,x_1,\cdots,x_{m-1})=(f_0,f_1,\cdots,f_{m-1})$,如果对任意的向量布尔函数$f_i$都包含所有的输入变量,则称该布尔函数具有完全性.

2.3 等价类

为了能够更好地研究向量布尔函数的性质,我们这里引入了等价类的概念. 相同等价类中向量布尔函数的密码学性质保持不变,引入该定义可以帮助我们更好地研究向量布尔函数. 其中仿射等价类和拓展比特置换等价类是最常用于向量布尔函数研究的等价类.

定义9 线性、仿射和比特置换等价:如果两个n bit到m bit布尔函数F和G,存在n bit到n bit的线性双射L_{in}和m bit到m bit的线性双射L_{out},使得

$$F=L_{\mathrm{out}}\circ G\circ L_{\mathrm{in}}$$

则称布尔函数F和G是线性等价的,同理我们可以定义仿射等价和比特置换等价. 显然这三个等价有包含关系,若两个布尔函数为比特置换等价,则同样满足仿射等价和线性等价;若两个布尔函数为线性等价,则同样满足仿射等价. 仿射等价可以视为线性等价的一个拓展,类似地我们可以将比特置换等价进行拓展,并由此得到拓展比特置换等价的概念,这也是本文中使用的等价类概念.

定义10 拓展比特置换等价:如果两个n bit到m bit的布尔函数F和G,存在n bit的比特置换P_{in}和m bit的比特置换P_{out},$\boldsymbol{\alpha}\in\mathbb{F}_2^n$和$\boldsymbol{\beta}\in\mathbb{F}_2$,使得对于所有的$\boldsymbol{x}\in\mathbb{F}_2^n$都有

$$F(\boldsymbol{x})=\left(P_{\mathrm{out}}\circ G\circ P_{\mathrm{in}}(\boldsymbol{x}\oplus\boldsymbol{\alpha})\right)\oplus\boldsymbol{\beta}$$

则称布尔函数F和G是拓展比特置换等价的,从定义可以看出拓展的比特置换等价同样是被仿射等价覆盖的.

在上述的所有等价类中,前文提到的(向量)布尔函数密码学性质,包括代数次数、线性度和差分均匀度都是保持相同的. 一个布尔函数等价类中字典序最小的布尔函数通常作为等价类的代表元布尔函数.

2.4 布尔函数的实现

为了能够更加直观地体现通过电路建模得到的布尔函数电路的延迟,研究中我们引入了复杂度的概念.

通常我们定义复杂度的概念会在一个特定的门电路集合$\mathcal{G}$上,集合中包含实现电路使用的所有逻辑门电路,这些门电路也被称为实现中使用的基. 为了保证集合的完备性,集合中给出的门电路基需要能够实现任意的布尔函数,例如$\mathcal{G}=\{\text{XOR, AND, INV}\}$.

定义 11 门电路复杂度:使用门电路集合$\mathcal{G}$中门电路实现函数需要的最少门电路数.

门电路复杂度是评估电路实现的一个重要指标. 尽管不同的工艺库中不同门电路的面积有较大的差异,但是门电路复杂度还是可以一定程度上反映函数的硬件实现中面积上的消耗. 然而由于没有考虑电路实现中存在的并行性,门电路复杂度并不能直观地反映函数实现的延迟. 因此我们引入了门电路深度复杂度的概念.

定义 12 门电路深度复杂度:使用门电路集合$\mathcal{G}$中门电路基实现一个函数,从函数任意输入到输出的最长路径的最小值.

门电路深度复杂度可以等价理解为S盒硬件实现中的关键路径,能够相对直观地反映一个电路实现的延迟. 虽然门电路深度复杂度能一定程度上反映延迟,但是在实际的硬件实现中,不同门电路,甚至相同门电路在不同的工艺库下的实现延迟都有较大的区别,因此实际评估电路实现的延迟时还需要具体考虑门电路的类型,这一点在接下来的构造和分类搜索的划分中也会有一定的体现.

3 低延迟布尔函数构造

本节将介绍如何使用低延迟的逻辑门电路来构造极致低延迟布尔函数. 使用低延迟门电路构造布尔函数的思想源自SPEEDY算法的6 bit S盒设计,其设计过程中选用逻辑门电路中硬件实现延迟较低NAND门,建立两层的树状门电路模型用于构造布尔函数及其等价类,通过搜索筛选出其中密码学性质较优的平衡布尔函数用于构造S盒. 鉴于Knežević等人[3]在2012年对分组密码算法在低延迟行为上的研究中提出:具有密码学强度3、4 bit S盒在延迟表现上相比更高输入比特的S盒更优. 本文选用了4 bit的布尔函数和S盒作为研究对象.

为了实现极低的延迟,本文采用了两层门电路构造布尔函数. 同时由于不同的工艺库中的门电路延迟不同,本文参照了表1中NanGate 45 nm OCL(Open Cell Library)和NanGate 15 nm OCL下不同门电路的扇入和延迟,选择其中适用于低延迟S盒设计的门电路. 从表1中可以看到:扇入为2时,NAND2门的延迟在两种不同的工艺库中都是最低的;扇入为3时,NanGate 45 nm OCL中OAI21门延迟最低,而NanGate 15 nm OCL中NAND3门延迟最低;扇入为4时,NanGate 45 nm OCL和NanGate 15 nm OCL中NAND4门和OAI22门分别有着最低的延迟.

不同于SPEEDY算法中在S盒设计时仅考虑单纯使用NAND门或OAI门构造布尔函数,本文在将OAI门和NAND门的混合使用加入搜索范围. 在延迟复杂度为2情况下构造布尔函数,同时考虑到实际中门电路的延迟不同,按照其门电路扇入数可以大致划分为3档,根据延迟水平由低到高分别为FAN-IN2, FAN-IN3, FAN-IN4. 特别地,两层模型中使用的逻辑门电路包含NAND2、NAND3、NAND4、OAI21和OAI22门电路,其中NAND2门属于FAN-IN2,NAND3和OAI21门属于FAN-IN3,NAND4和OAI22门属于FAN-IN4. 考虑到NAND4门和OAI22门电路面积更大且延迟更高,此处仅考虑二者用于模型第二层的情况. 需要注意的是,模型中使用门电路的扇出均为1. 遵从上述的限制,本文建立了如图1所示的基于低延迟门电路和双层树型结构的布尔函数模型.

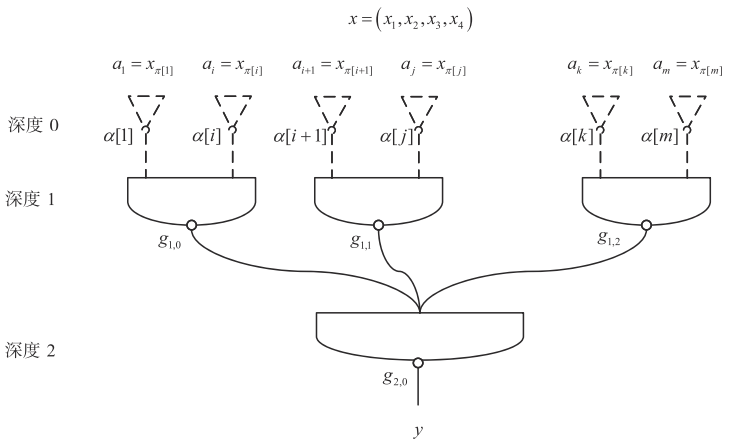

图1 双层树型结构布尔函数模型

表1　NanGate 15 nm和NanGate 45 nm OCL下常用门电路延迟　　单位:ps

输入	门电路类型	输出	NanGate15 nm OCL 延迟	NanGate 45 nm OCL 延迟
x_0	INV	$\neg x_0$	1.580	22.048
	BUF	x_0	3.068	33.557
x_0x_1	NAND2	$\neg(x_0\wedge x_1)$	2.031	27.886
	NOR2	$\neg(x_0\vee x_1)$	2.554	40.650
	AND2	$x_0\wedge x_1$	3.580	40.171
	OR2	$x_0\vee x_1$	3.644	56.414
	XOR2	$x_0\oplus x_1$	5.268	73.019
	XNOR2	$\neg(x_0\oplus x_1)$	6.788	57.604
$x_0x_1x_2$	NAND3	$\neg(x_0\wedge x_1\wedge x_2)$	2.361	34.767
	OAI21	$\neg((x_0\vee x_1)\wedge x_2)$	2.830	32.651
	AOI21	$\neg((x_0\wedge x_1)\vee x_2)$	3.394	51.619
	NOR3	$\neg(x_0\vee x_1\vee x_2)$	3.788	61.543
	AND3	$x_0\wedge x_1\wedge x_2$	5.496	51.869
	OR3	$x_0\vee x_1\vee x_2$	5.862	85.840
	MUX2	$(\neg x_0\wedge x_1)\vee(x_0\wedge x_2)$	6.133	75.175
$x_0x_1x_2x_3$	OAI22	$\neg((x_0\vee x_1)\wedge(x_2\vee x_3))$	3.776	54.596
	AOI22	$\neg((x_0\wedge x_1)\vee(x_2\wedge x_3))$	4.070	57.255
	NAND4	$\neg(x_0\wedge x_1\wedge x_2\wedge x_3)$	4.659	44.487
	NOR4	$\neg(x_0\vee x_1\vee x_2\vee x_3)$	5.250	91.313
	AND4	$x_0\wedge x_1\wedge x_2\wedge x_3$	7.125	65.492
	OR4	$x_0\vee x_1\vee x_2\vee x_3$	7.683	118.592

假设布尔函数的输入变量为$\boldsymbol{x}\in\mathbb{F}_2^4$,其中$a_i\in\mathbb{F}_2$表示对应的第一层输入比特的值,$\boldsymbol{\alpha}\in\mathbb{F}_2$表示$a_i$是否经过一个INV门,若$\alpha[i]=1$,则第一层的输入为$\bar{a}_i$. $\pi[i]\in\mathbb{Z}_4$表示a_i对应的输入比特,即$a_i=x_{\pi[i]}$. 通过将不同的门电路映射到对应的值,我们可以用$g_{i,j}$表示第i层第j个门电路的类型,此处我们令NAND2、NAND3、OAI21、NAND4和OAI22门分别对应$g_{i,j}$为0,1,2,3,4. 接着我们可以用$\boldsymbol{G}\in\mathbb{F}_2^{n+1}$表示结构中所有门电路的类型,其中$n$为第一层门电路的数量,$n$的值由第二层门电路的扇入数决定,其中$\boldsymbol{G}$具体可以表示为$\boldsymbol{G}=(g_{1,0},\cdots,g_{1,n},g_{2,0})$.

举一个例子,假设第二层门电路为NAND4门,第一层门电路分别为NAND3, OAI21, NAND3和OAI21, 此时$\boldsymbol{G}=(1,2,1,2,3)$. 若门电路输入为$(x_0,\bar{x}_2,x_3,\bar{x}_1,x_1,\bar{x}_2,x_2,x_3,\bar{x}_1,x_0,x_1,x_3,)$, 则我们可以得到对应的$\boldsymbol{\alpha}=(0,1,0,1,0,1,0,0,1,0,0,0)$和$\boldsymbol{G}=(0,2,3,1,1,2,2,3,1,0,1,3)$.

定义上述符号后,我们可以用$(\boldsymbol{G},\boldsymbol{\alpha},\boldsymbol{\pi})$来唯一确定一个布尔函数的电路实现,接着再考虑$\boldsymbol{G}$中门电路类型的选择. 首先可以确认的是由于NAND2门作为第二层的结构过于简单无法提供必要安全性,所以仅考虑使用扇入为3的门电路作为第二层的可能选择. 由此得到第二层的门电路可能的选择有NAND4、NAND3、OAI22和OAI21门;其次考虑第一层门电路的选择,排除多次使用可能导致面积和延迟增大的NAND4和OAI22门,可用的选择有NAND2、NAND3、OAI21门. 进而根据不同扇入门电路延迟的区别,可以按延迟高低的顺序去划分出以下四种情况,对应的模型如图2所示.

情况1:延迟等价于FAN-IN3 + FAN-IN2.

情况2:延迟等价于FAN-IN3 + FAN-IN3.

情况3:延迟等价于FAN-IN4 + FAN-IN2.

情况4:延迟等价于FAN-IN4 + FAN-IN3.

接着根据上述四种情况的结构分类搜索得到所有的情况,可以得到对应的布尔函数. 实际上,如果仅使用NAND门和NOT门并将本文的模型拓展,该模型可以覆盖所有的布尔函数,即使用仅包含NAND(与非)门的逻辑电路可以表示任意的布尔函数. 根据De Morgan定律,对于任意布尔函数A和B,

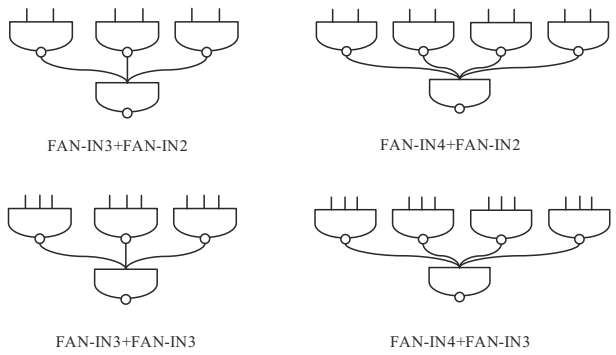

图2 不同延迟水平的模型

都有 $A\,\mathrm{AND}\,B=\mathrm{NOT}\big(\mathrm{NAND}(A,B)\big)$ 和 $A\,\mathrm{OR}\,B=\mathrm{NAND}(\mathrm{NOT}\,A,\mathrm{NOT}\,B)$,由于AND门、OR门和NOT门是一个完备的逻辑门集合,即能够实现任意的布尔函数,利用上述两个规则可以将任意布尔函数用AND门和OR门实现,然后转化为使用NAND门的实现.由此我们可以证明通过NAND门和NOT门组成的门集具有完备性,可以用该门集构造任意的布尔函数.

需要注意的是,尽管可以使用该模型来表示任意的布尔函数,但这并不意味着是最优的或最简洁的表示.对于特定的布尔函数,可能存在更简单或更高效的逻辑电路表示方式.同时,因为这里仅使用了两层NAND门,所以通过上述方法构造的结果为拥有低延迟实现的部分布尔函数,剩余的布尔函数可能需要有更多层数的类似结构或是拥有更多扇入扇出的门电路来实现.

以情况4为例,假设第二层为NAND4门,第一层包含NAND3、NAND2和OAI21门,这种情况下 $\boldsymbol{G}$ 有 3^4 种情况,$\boldsymbol{\alpha}$ 有平均 $2^{10.7}$ 种情况,$\boldsymbol{\pi}$ 有平均 $4^{10.7}$ 种情况.若是遍历所有的情况,则搜索的复杂度大约为 $2^{6.3+10.7+21.4}=2^{48.4}$.如果输入比特大于4、层数或者门电路类型增加,想要再使用这种方法搜索低延迟的S盒在实际计算复杂度上难以实现,因此需要在门电路类型上有更多的限制.

在搜索得到所有的布尔函数结果后,需要考虑布尔函数是否满足密码学性质的要求.接着对搜索得到的布尔函数以平衡性、完全性和线性度为指标进行筛选,从而得到满足下列密码学需求的布尔函数:

(1)平衡性:布尔函数真值表01数目相同.

(2)完全性:布尔函数是否由全部的S盒输入比特决定.

(3)线性度:布尔函数对应线性度应小于设定值.

需要注意的是,布尔函数的平衡性和线性度都可以通过预计算存储在哈希表中,之后只需要通过查表即可得到对应布尔函数的指标从而进行筛选.

出于后续构造向量布尔函数的需求,对搜索得到的布尔函数需要按照等价类进行划分,本文采用了拓展比特置换等价类对搜索得到满足需求的布尔函数进行了划分.4 bit布尔函数共有65 536个,以拓展比特置换等价类进行划分可以得到402个拓展比特置换等价类.基于上述方法,以线性度为8具有完全性的平衡布尔函数为标准,在Ubuntu 20.04.5操作系统,处理器为Intel(R) Xeon(R) Gold 6248 CPU @ 2.50 GHz,内存为1 TB,在VS code中使用C++编程语言实现该过程,得到如表2所示的实验结果.

表2 低延迟布尔函数构造结果

延迟水平	第一层门电路	第二层门电路	布尔函数总数	布尔函数比特置换等价类
FAN-IN3+FAN-IN2	NAND2	NAND3	192	1
	NAND2	OAI21	0	0
FAN-IN3+FAN-IN3	NAND2\\NAND3	NAND3	1 248	7
	NAND2\\NAND3\\OAI21	NAND3	2 984	17
	NAND2\\NAND3\\OAI21	OAI21	3 349	19
FAN-IN4+FAN-IN2	NAND2	NAND4	200	2
	NAND2	OAI22	0	0
FAN-IN4+FAN-IN3	NAND2\\NAND3	NAND4	7 052	38
	NAND2\\NAND3\\OAI21	OAI22	7 308	39

表2中可以看到,NAND3+NAND2和NAND4+NAND2延迟水平下,若第二层门电路为OAI门,构造的布尔函数均无法满足平衡性的需求.第二层为NAND3门、第一层为NAND2门时,搜索可以得到192个布尔函数,但是均为1个比特置换等价类,原因是该结构搜索得到的布尔函数都是比特置换等价的,在搜索过程中只是改变了输入变量的位置和是否经过一个非门,相当于对布尔函数的输入变量的定义域作了比特置换操作和异或运算.

搜索得到的布尔函数均满足平衡性、输出依赖于所有输入和线性度低于8的要求.进一步研究搜索得到的布尔函数性质发现:虽然4 bit布尔函数能够达到的线性度最低为4,但是使用这类结构能够构造的平衡布尔函数线性度的最佳结果为8.可以看到当第二层为

NAND3门,第一层为NAND2、NAND3门时,可以得到1 248个布尔函数,7个比特置换等价类;当第二层为NAND3门,第一层为NAND3、NAND2和OAI21门时,可以得到2 984个布尔函数,17个比特置换等价类;第二层为OAI21门,第一层为NAND3、NAND2和OAI21门,可以得到3 349个布尔函数,19个比特置换等价类;第一层为NAN4门,第二层为NAND3、NAND2门,可以得到7 052个布尔函数,38个比特置换等价类;第一层为OAI22门,第二层为NAND3、NAND2、OAI21门,可以得到7 308个布尔函数,39个比特置换等价类.

4　低延迟S盒构造

本节将介绍如何使用上一节搜索得到的布尔函数来构造满足密码学性质的极致低延迟4 bit S盒,并基于构造得到的S盒搜索双向低延迟S盒.

4.1　低延迟S盒的构造方法

基于上一节找到的布尔函数和代表元布尔函数集合,我们希望将其作为分量布尔函数构造出低延迟的S盒. 与构造布尔函数的方法相同,我们希望最终得到的S盒能够满足一定的密码学性质,具有良好的差分、线性性质,此处应引入以下的定理.

定理1[19]　对于一个线性度为l的n bit双射S盒S,每个组件布尔函数是平衡且线性度至多为l.

基于上述定理,应用布尔函数及其等价类来构造向量布尔函数的方法可以用如下流程描述:首先从代表元布尔函数的集合中选择一个布尔函数f_0,接着从布尔函数集合中取一个布尔函数f_1,检验$f_0 \oplus f_1$是否满足平衡性与线性度要求;若满足要求则再从布尔函数集合中取布尔函数f_2,检验$f_0 \oplus f_2$、$f_1 \oplus f_2$、$f_0 \oplus f_1 \oplus f_2$是否满足平衡性和线性度要求;同理可以选取$f_3$,由此可以选定所有的向量布尔函数用于构建S盒,最后再对其的差分均匀度进行检验,即可筛选出满足条件的S盒.

然而经过实验测试,上述方法在仅使用NAND3门、NAND2门,情况较少的时候可以在短时间内得到结果,当门电路的选择增加或是代表布尔函数过多时,构建S盒需要的时间将会显著增长. 因此本文使用如下构建S盒的方法:

(1)首先从代表布尔函数中选取一个布尔函数f_0^*,并计算其比特置换等价布尔函数的集合D.

(2)遍历D中的所有f_1,若$f_0^* \oplus f_1$满足平衡性与线性度要求,则将其加入集合D_1.

(3)接着遍历D_1中的f_1,并从D_1中任取一个f_2,若加入f_2后,所有的组件函数均满足平衡性与线性度要求,则将f_2加入集合D_2.

(4)重复直到找到所有的坐标布尔函数,并验证S盒的差分均匀度是否满足条件,若满足条件则将S盒加入候选低延迟S盒集合.

该方法减少了每轮向量布尔函数选择的空间,但是增加了一定的存储空间,需要存储上一轮得到的满足条件的所有布尔函数,在时间和空间的消耗上进行了一定的权衡.

4.2　低延迟S盒的构造结果

在Ubuntu 20.04.5操作系统,处理器为Intel(R) Xeon(R) Gold 6248 CPU @ 2.50 GHz,内存为1 TB,在VS code中使用C++编程语言实现S盒构造和筛选的过程,得到如表3所示的实验结果.

表3　低延迟S盒构造结果

延迟水平	第一层门电路	第二层门电路	S盒总数
FAN-IN3+FAN-IN3	NAND2\\NAND3	NAND3	357 863
	NAND2\\NAND3\\OAI21	NAND3	2 947 975
	NAND2\\NAND3\\OAI21	OAI21	4 208 403
FAN-IN4+FAN-IN3	NAND2\\NAND3	NAND4	244 283 818
	NAND2\\NAND3\\OAI21	OAI22	210 628 146

以第一层使用NAND2和NAND3门,第二层使用NAND3门的S盒为例,以线性度8和差分均匀度4为标准构建S盒,我们得到了357 863个结果. 接着计算得到所有S盒的代数次数,可以发现用这种方法得到的S盒的代数次数均为3. 考虑到用该方法构造的S盒关键路径相同,延迟相同,进一步考虑硬件实现面积上减少消耗,选取构造过程中使用最少NAND3门的S盒,发现共有192个S盒. 再对其进行比特置换等价分类,发现共有4类S盒,从4类S盒中选出固定点数目最少的S盒作为代表,结果如表4所示.

表4　非线性度8和差分均匀度4的单向低延迟S盒构造结果

S盒	延迟/ns	面积/GE
3, 0, 6, 2, 5, 4, 15, 14, 10, 8, 7, 9, 1, 12, 13, 11	0.12	11.17
1, 3, 4, 7, 2, 0, 8, 12, 14, 11, 5, 15, 6, 10, 13, 9	0.12	11.17
3, 2, 7, 1, 6, 4, 15, 13, 11, 10, 5, 8, 0, 14, 12, 9	0.12	11.17

在NanGate 45 nm工艺库下,使用Synopsys Design Compiler综合和XilinxISE 14.7模拟对S盒进行测试,得到上述S盒的延迟均为0.12 ns,面积分别为11.17 GE与11.70 GE. 以第一个S盒为例,下文称其为LLSB (Low Latency S-Box). 我们可以查表得到LLSB的分量布尔函数低延迟实现如表5所示.

S盒的门电路树型结构如图3所示.

表5 低延迟S盒的门电路实现

布尔函数序号	门电路类型	输入
Boolfunction1	NAND2	x_1,x_2
	NAND3	$\neg x_1,\neg x_2,x_3$
	NAND2	x_0,x_3
Boolfunction2	NAND2	$x_2,\neg x_3$
	NAND3	$x_0,\neg x_1,x_2$
	NAND2	$\neg x_0,x_1$
Boolfunction3	NAND2	$x_1,\neg x_3$
	NAND3	x_0,x_1,x_2
	NAND2	$\neg x_0,\neg x_2$
Boolfunction4	NAND2	x_1,x_3
	NAND3	$\neg x_0,\neg x_1,\neg x_3$
	NAND2	$\neg x_0,x_2$

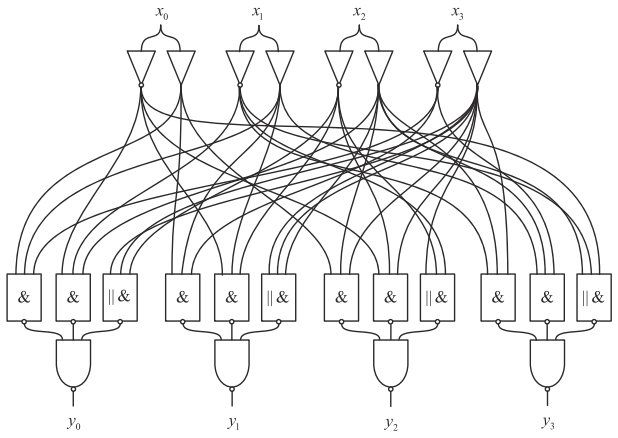

图3 低延迟S盒的门电路树型结构

5 双向低延迟S盒的构造

在低延迟密码设计的过程中,设计者们往往都会考虑利用拥有α反射特性的结构来节约密码算法在硬件上全展开实现需要的面积消耗,因此轮函数需要同时实现正向和逆向S盒,使得逆向S盒的实现成本也至关重要.但是在低延迟密码SPEEDY设计中并没有使用这种结构,同时其设计过程中仅考虑正向S盒的延迟水平,放弃考虑解密过程以及逆S盒实现需要的代价.这也导致使用其方法构造低延迟的S盒在实现过程中加密延迟极低,但是解密延迟相对偏高.若我们能够在正向保证低延迟的情况下尽可能使其逆向S盒同样低延迟,那么显然在延迟性质上这种S盒是更加有优势的,我们将具有这类性质的S盒称为具有双向低延迟性质的S盒.

因此本文在低延迟单向S盒搜索完毕后增加S盒集合与逆S盒集合的匹配过程,从而寻找双向低延迟S盒.我们首先计算搜索得到S盒集合中所有S盒对应的逆S盒,将S盒与逆S盒集合以分量布尔函数编码为对应值后存入对应的无序集合容器中,接着匹配两个无序集合中的重复元素.该过程中由于使用的容器为无序集合,搜索指定元素的复杂度为$O(1)$,因此整个匹配过程的复杂度为$O(n)$.

以第一层使用NAND2和NAND3门、第二层使用NAND3门的S盒为例,根据上一节搜索的结果,我们得到357 863个S盒后,进一步使用上述方法筛选逆S盒同样在低延迟S盒集合中的结果.我们可以得到如下表所示的三个S盒,同样考虑其S盒实现中使用的NAND3门数量和固定点,得到如表6所示的结果.

表6 部分双向低延迟S盒构造结果

S盒	固定点数	硬件实现中NAND3门数
4, 3, 5, 1, 0, 2, 13, 10, 8, 9, 7, 11, 12, 6, 15, 14	4	6
3, 4, 2, 0, 1, 5, 10, 13, 9, 8, 6, 12, 11, 7, 14, 15	4	6
2, 1, 0, 3, 4, 5, 8, 10, 6, 14, 7, 15, 12, 13, 9, 11	6	7

可以发现仅使用NAND3门和NAND2门构造的低延迟S盒内部匹配,虽然可以实现具有双向的低延迟性质的S盒,但是得到的S盒在固定点的数量上过多,无法满足我们对S盒安全性的要求.

因此进一步考虑在第一层加入OAI21门的结果,根据上一小节的结果我们得到了包含2 947 976个单向低延迟S盒的集合,接着应用上述集合内部匹配的方法我们找到了3 089个S盒,能够在低延迟S盒集合中找到其对应的逆S盒,先筛选其中满足无固定点的S盒,再对得到的S盒进行比特置换等价分类,可以筛选得到118个比特置换等价类.对于第二层使用OAI21门的情况,从4 208 403个单向低延迟S盒中可以得到528个双向低延迟S盒的比特置换等价类.对于第二层使用NAND4和OAI22门电路的低延迟S盒,我们能够找到双向低延迟S盒,但是想要计算其比特置换等价类需要的时间过高,且得到S盒的延迟水平要低于前两类,故此处不做进一步的搜索.

在Ubuntu 20.04.5操作系统,处理器为Intel(R) Xeon(R) Gold 6248 CPU @ 2.50 GHz,内存为1 TB,在VS code中使用C++编程语言运行实现.针对不同的延迟水平,使用文中提出的双向低延迟S盒构造方法,进一步对得到的S盒搜索后我们得到了如表7的搜索结果.

由于NAND3+NAND3延迟水平最低,因此选取第一层为NAND3、OAI21门,第二层为NAND3门的S盒中的一个S盒为例,下文称其为BLLSB (Bidirectional Low Latency S-Box).下面给出了其正向和逆向低延迟实现以及其正向实现对应的电路图,其中正向S盒的低延迟

表7　双向低延迟S盒的搜索结果

延迟水平	第一层门电路	第二层门电路	S盒总数
FAN-IN3+FAN-IN3	NAND2\\NAND3	NAND3	4
	NAND2\\NAND3\\OAI21	NAND3	3 089
	NAND2\\NAND3\\OAI21	OAI21	1 403
FAN-IN4+FAN-IN3	NAND2\\NAND3	NAND4	359 586
	NAND2\\NAND3\\OAI21	OAI22	558 859

实现如表8所示,S盒的逆向低延迟实现如表9所示,S盒对应的正向门电路实现如图4所示.

表8　双向低延迟S盒的正向低延迟实现

布尔函数序号	门电路类型	输入
Boolfunction5	NAND3	$x_0,x_1,\neg x_3$
	NAND3	$\neg x_0,\neg x_1,x_2$
	OAI21	$x_3,x_3,\neg x_2$
Boolfunction6	NAND3	$x_0,x_1,\neg x_3$
	NAND3	$\neg x_0,x_1,\neg x_2$
	OAI21	x_3,x_3,x_2
Boolfunction7	NAND3	x_0,x_2,x_3
	NAND3	$\neg x_0,\neg x_2,x_3$
	OAI21	$x_3,\neg x_2,\neg x_1$
Boolfunction8	NAND3	$\neg x_1,x_2,x_3$
	NAND3	$x_1,x_2,\neg x_3$
	OAI21	$x_2,\neg x_1,\neg x_0$

表9　双向低延迟S盒的逆向低延迟实现

布尔函数序号	门电路类型	输入
Boolfunction9	NAND3	$\neg x_0,\neg x_2,x_3$
	NAND3	$x_0,\neg x_2,\neg x_3$
	OAI21	x_3,x_2,x_1
Boolfunction10	NAND3	$x_1,x_2,\neg x_3$
	NAND3	$\neg x_1,\neg x_2,\neg x_3$
	OAI21	$x_2,\neg x_1,x_0$
Boolfunction11	NAND3	$\neg x_0,\neg x_1,x_3$
	NAND3	$x_0,\neg x_1,\neg x_3$
	OAI21	$\neg x_1,\neg x_0,x_2$
Boolfunction12	NAND3	$\neg x_1,x_2,x_3$
	NAND3	$x_1,x_2,\neg x_3$
	OAI21	$\neg x_2,\neg x_2,\neg x_0$

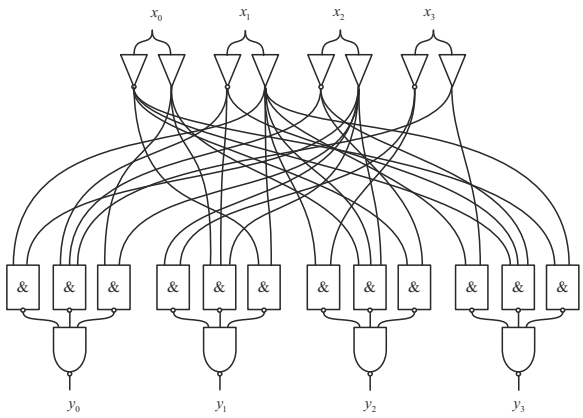

图4　双向低延迟S盒的正向门电路树型结构

6　实验结果与分析

在NanGate 45 nm工艺库下,使用Synopsys Design Compiler综合和XilinxISE 14.7模拟本文构造的低延迟S盒与轻量级分组密码算法中使用S盒的延迟和面积,其中单向低延迟S盒为第4节中得到的四个S盒中的第一个,得到的实验测试结果如表10所示.

表10　低延迟S盒与轻量级密码S盒硬件实现测试结果

密码算法		延迟/ns	面积/GE
PRINCE	正向	0.18	13.57
	逆向	0.18	15.96
MANTIS	正向	0.15	11.97
	逆向	0.15	11.97
Orthros	正向	0.13	13.83
	逆向	0.19	15.69
Midori1	正向	0.12	10.91
	逆向	0.12	10.91
Midori2	正向	0.12	12.24
	逆向	0.12	12.24
单向低延迟S盒LLSB	正向	0.12	11.17
	逆向	—	—
双向低延迟S盒BLLSB	正向	0.15	14.63
	逆向	0.15	14.63

可以看到本文构造的S盒与其他低延迟分组密码算法PRINCE和MANTIS中使用的4 bit S盒相比,在延迟和实现面积上都有所提升. 其中与PRINCE的S盒相比,仅考虑S盒的正向实现时,第4节中构造的S盒LLSB在硬件实现延迟上降低了33%,实现面积上降低了17.69%;与MANTIS构造的S盒相比,由于其S盒是

自逆的,第4节中构造的S盒LLSB在硬件实现延迟上降低了20%,实现面积上降低了6.68%;与Orthros构造的S盒相比,仅考虑正向实现,LLSB在硬件实现延迟上降低了7.69%,实现面积上降低了19.2%;与Midori的S盒相比,虽然在延迟上没有显著提升,但本文的方法提供了更多低延迟S盒的选择. 第5节中构造的双向低延迟S盒BLLSB虽然与MANTIS相比延迟上没有更多提升,但是与PRINCE相比在正向和逆向的延迟水平上均有一定提升.

7 结论

本文首先选用低延迟门电路和两层树型结构建模得到不同延迟水平下具有一定密码学性质的平衡4 bit布尔函数及其拓展比特置换等价类;接着采用逐步构造的方法将布尔函数作为向量布尔函数的分量布尔函数构造出满足需求的低延迟S盒,并进一步给出其中延迟性质和硬件实现面积上有优势的S盒;此外我们还对搜索得到的低延迟S盒与对应的低延迟逆S盒匹配搜索取交集,给出了不同延迟水平下有双向低延迟性质的S盒搜索结果及实例;最后本文在NanGate 45 nm工艺库下,使用Synopsys Design Compiler综合和XilinxISE 14.7模拟,将本文得到的超低延迟4 bit S盒与其他低延迟分组密码中使用的4 bit S盒比较,本文得到的低延迟S盒在延迟水平上相较MANTIS降低了20%,与PRINCE相比降低了33%,在硬件实现面积上相较MANTIS减少了6.68%,与PRINCE相比减少了17.69%.

致谢 感谢吴文玲老师给本文提出的参考意见.

参考文献

[1] COSTAN V, DEVADAS S. Intel SGX explained[EB/OL]. (2017-02-21)[2024-03-10]. https://eprint.iacr.org/2016/086.pdf.

[2] KAPLAN D, POWELL J, WOLLER T. AMD memory encryption[EB/OL].(2021-10-18)[2024-03-10].https://www.amd.com/content/dam/amd/en/documents/epyc-busin-essdocs/white-papers/memory-encryption-white-paper.pdf.

[3] KNEŽEVIĆ M, NIKOV V, ROMBOUTS P. Low-latency encryption—is "lightweight = light + wait"?[M]//Lecture Notes in Computer Science. Berlin: Springer, 2012: 426-446.

[4] BORGHOFF J, CANTEAUT A, GÜNEYSU T, et al. PRINCE—A low-latency block cipher for pervasive computing applications[M]//Lecture Notes in Computer Science. Berlin: Springer, 2012: 208-225.

[5] BOŽILOV D, EICHLSEDER M, KNEŽEVIĆ M, et al. PRINCEv2: More security for (almost) no overhead[M]//Lecture Notes in Computer Science. Cham: Springer International Publishing, 2021: 483-511.

[6] BANIK S, BOGDANOV A, ISOBE T, et al. Midori: A block cipher for low energy[M]//Lecture Notes in Computer Science. Berlin: Springer, 2015: 411-436.

[7] BEIERLE C, JEAN J, KÖLBL S, et al. The SKINNY family of block ciphers and its low-latency variant MANTIS[M]//Lecture Notes in Computer Science. Berlin: Springer, 2016: 123-153.

[8] AVANZI R. The QARMA block cipher family. almost MDS matrices over rings with zero divisors, nearly symmetric even-mansour constructions with non-involutory central rounds, and search heuristics for low-latency S-boxes[J]. IACR Transactions on Symmetric Cryptology, 2017: 4-44.

[9] LEANDER G, MOOS T, MORADI A, et al. The speedy family of block ciphers—Engineering an ultra low-latency cipher from gate level for secure processor architectures[EB/OL]. (2021-07-22) [2024-03-10]. https://eprint. iacr.org/2021/960.pdf.

[10] BANIK S, ISOBE T, LIU F K, et al. Orthros: A low-latency PRF[EB/OL]. (2021-03-27) [2024-03-10]. https://eprint.iacr.org/2021/390.pdf.

[11] BIRYUKOV A, DE CANNIÈRE C, BRAEKEN A, et al. A toolbox for cryptanalysis: Linear and affine equivalence algorithms[M]//Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003: 33-50.

[12] LEANDER G, POSCHMANN A. On the Classification of 4 Bit S-Boxes[M]//Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007: 159-176.

[13] ZHANG W T, BAO Z Z, RIJMEN V, et al. A new classification of 4-bit optimal S-boxes and its application to PRESENT, RECTANGLE and SPONGENT[C]//Fast Software Encryption. Berlin: Springer, 2015: 494-515.

[14] BOŽILOV D, BILGIN B, ALI SAHIN H . A note on 5-bit quadratic permutations' classification[J]. IACR Transactions on Symmetric Cryptology, 2017, 2017(1): 398-404.

[15] DE MEYER L, BILGIN B. Classification of balanced quadratic functions[J]. IACR Transactions on Symmetric Cryptology, 2019: 169-192.

[16] 赵耿, 马英杰, 陈磊, 等. 基于扰动时空混沌系统的动态S盒设计[J]. 电子学报, 2022, 50(8): 2037-2042.
ZHAO G, MA Y J, CHEN L, et al. Design of dynamic S-box based on perturbed spatiotemporal chaotic system[J]. Acta Electronica Sinica, 2022, 50(8): 2037-2042. (in Chinese)

[17] 柴进晋, 吴暄. 一种抗错误注入攻击的S盒的构造[J]. 电子学报, 2023, 51(12): 3422-3430.
CHAI J J, WU X. Construction of fault injection attacks resistant S-boxes[J]. Acta Electronica Sinica, 2023, 51(12): 3422-3430. (in Chinese)

[18] KO S. Optimizing S-Box implementations for several criteria using SAT solvers[M]//Lecture Notes in Computer Science. Berlin: Springer, 2016: 140-160.

[19] RASOOLZADEH S. Low-latency boolean functions and bijective S-boxes[J]. IACR Transactions on Symmetric Cryptology, 2022(3): 403-447.

作者简介

吴瑞宸　男, 1999年8月出生于浙江省温州市. 现为中国科学院软件研究所硕士研究生. 主要研究方向为分组密码设计与分析.
E-mail: ruichen2021@iscas.ac.cn

张　蕾　女, 1982年出生于吉林省. 2010年毕业于中国科学院软件研究所. 现为中国科学院软件研究所副研究员. 主要研究方向为分组密码设计与分析.
E-mail: zhanglei@tca.iscas.ac.cn