**班级**

**学号**

**本科毕业设计（论文）**

**外文资料翻译**

**毕业设计题目**

**外文资料题目**

**学 院**

**专 业**

**学 生 姓 名**  龚若涵

**指导教师姓名**  王子龙

# S盒的检测、实现和生成平台-PEIGEN

**摘要：**本文提出了一个名为PEIGEN的平台，用于评估S盒的安全性，寻找有效的软件/硬件实现，并生成加密的S盒。经过几十年的持续发展，S盒在安全要求和软/硬件性能的设计标准方面都在不断改变。PEIGEN旨在成为一个覆盖文献中出现的S盒设计标准的综合检查清单的平台。为此，本文中首先对安全要求进行深入调查，然后对现有的S盒相关工具进行了全面的比较，最后提出了平台PEIGEN。 综述部分是为了给S盒的理论研究提供系统的参考。该平台是为S盒的实验研究和实际使用提供一个辅助工具。PEIGEN不仅整合了现有工具中的大部分功能，而且还配备了评估新的安全属性的功能，从多个方面提高了优化实现S盒的搜索算法效率。通过对该平台的使用，在安全符号之间以及在现有的对称密码加密原语中使用的S盒上进行了具有意义的考察。PEIGEN将成为一个开放的平台，并欢迎各方的贡献来助力社区促进S盒的研究和使用。

关键词：S盒 调研 设计准则 实现准则 新平台

1、引言

substitution-box，简称S盒，通常用于设计对称密码原语以提供非线性。一般来说，S盒是一个定义在上的非线性映射，它将一个n比特的值作为输入，输出一个m比特的值。 例如，数据加密标准（DES）的S盒是（n=6，m=4）的映射，高级加密标准（AES）的S盒是（n=m=8）的置换。S盒有两个关键方面：安全强度和软件/硬件的性能。在许多情况下，这两个优点是冲突的，因此要进行权衡以符合整个设计的优先次序。

S盒的安全要求，取决于整个原语的设计策略，并随着时间的推移而发展变化。每个安全要求都对应于抵抗某些密码攻击的目标，例如，S盒的高代数度可以作为抵抗代数攻击的指标，如积分攻击。随着新的攻击被设计出来，新的属性被添加到S盒的安全要求池中，例如，针对PRINTcipher [KLPR10]和Midori[BBI+15]的不变子空间攻击[LAAZ11, GJN+16]引发了对S盒和圆常数组合的研究/约束，这在2011年前是不存在的。同时，并非每个设计都必须满足所有的安全要求，因为有些攻击可以由其他设计组件来抵御，例如，圆常数可以用来打破对称性和抵御滑动攻击。因此，在S盒所需的安全属性子集和整个原语的其他组件的复杂性之间存在着一种权衡。

S盒的实现方面包括软件和硬件。根据平台和可用资源的不同，在软件中实现S盒有以下几种方法。一种常见的方法是查表法，通过穷尽所有可能的输入值并将输出存储在一个表中，预先计算出一个查找表（Look-Up Table , LUT）。随后通过访问该表来运用S盒，通常以输入值为索引。LUT方法会受到缓存定时攻击[Ber05]。为了抵抗该攻击，有一些（接近）恒定时间的实现方法，如代数实现和位片实现。两者都以代数形式表达S盒，因此执行时间与输入值无关。不同的是，位片实现通常采取多个并行输入，并通过利用长寄存器，如数据流单指令多数据扩展指令集（SSE）寄存器，同时处理所有的输入，这样可以显著提高每个输入的相对性能。

根据目标用途的不同，最常见的集成电路（IC）分为专用集成电路（ASIC）和现场可编程门阵列（FPGA）。在这两种集成电路上，S盒的实现和性能会有很大不同。在这两种类型的集成电路上，也有不同的优化方式，这将导致非常不一样的性能。例如，在ASIC和FPGA上，存在所谓的串行化实现，旨在占用最小的面积，而深度优化实现通常占用相对较大的面积，但会产生较高的频率和吞吐量。

为了在安全性和性能之间做出更好的权衡，有一个研究方向是开发寻找最佳实现的工具。给定一个S盒，Osvik [Osv00]试图找到在CPU周期方面最优化的软件实现。给定一个S盒，Guo等人[GJN+16]提出了一种方法来寻找在ASIC中以电路深度优化的实现。Jean等人[JPST17]给定了一个S盒和每个单元操作的成本，建立了一个名为LIGHTER的工具来寻找ASIC中小面积的实现或软件中指令数量较少的实现。表5中列出了更多的相关工作。

**本文贡献：**我们的社区不断努力研究S盒，希望提供最佳的设计选择，以产生不仅强大到足以抵御攻击，而且高效实际的密码。这些努力包括在偶尔的突发攻击之后为补救而发挥的作用，以及为寻求最佳实例以丰富候选设计组合而做出的努力。在本文中，我们结合了这些努力所取得的成果。本文首先遵循了为S盒的开发设计标准的研究路线。社区在这一领域已经取得了丰富的成果，一些早期研究的结果已被证明是基本理论，而一些新提出的概念则需要进一步发展。我们会提一个全面的展览，其主线是遵循攻击的路线。许多已知的结果提出或暗示了避免每一类攻击的标准。一些标准通常可以适用于任何基于S盒的设计，而另一些标准则假定是一种特殊类型的圆形函数结构。本文将这两种类型都考虑在内，为设计者形成一个全面的检查清单。除了安全性之外，我们还考虑了S盒准则的有效性。另外，我们还考虑到S盒的等价类和不同类之间的联系。这将有助于设计者确定满足特定标准的S盒的搜索空间。以前关于S盒的大量研究使我们的展示的内容相当多。然而，将漫长的S盒设计历史中出现的许多标准整合，对未来的S盒设计者和研究者来说是很有用。

在本文的后半部分，我们将遵循开发用于评估和实现S盒的工具这一研究路线。首先，我们全面地调查了工具开发的现状。该过程中发现：不同的工具使用不同的语言编写，具有不同的界面，其中部分工具是不公开的。因为每个工具都专注于其目标标准，所以我们需要使用/实现多个工具来评估多个标准，这是非常重要的。特别是，安全和实现方这两方面往往很难同时考虑。这促使我们提出了一个名为PEIGEN的平台，用于评估安全属性，为给定的S盒寻找最佳实现，并在给定的安全和性能要求下生成所有合适的S盒。

PEIGEN是在现有工具的基础之上建立的。经过综合比较，我们决定使用Jean等人[JPST17]提出的LIGHTER的实现模型来构建PEIGEN。更明确地说，PEIGEN是在LIGHTER的一个子模块的基础上建立的，该子模块用于寻找在4位双射S盒的位片门复杂度（Bitslice Gate Complexity, BGC）、门等效复杂度（Gate Equivalent Complexity, GEC）和乘法复杂度（Multiplicative Complexity , MC）方面良好的软件/硬件实现。PEIGEN继承了LIGHTER提供的所有功能，同时利用算法层面的优化提高了搜索效率，如组合和串联方法以及预计算机制。因此，PEIGEN对于大型的S盒来说，效率更高。除了支持4位的S盒，PEIGEN还支持n位(3≤n≤8)的S盒，但仍然只适用于寻找3位和4位S盒的实现）。

最重要的是，与LIGHTER相比Peigen的功能更全面：除了在BGC、GEC和MC方面能找到好的实现，Peigen还能找到在电路深度（Depth）方面好的实现；除了找到好的实现，Peigen还能评估与安全有关的属性并识别等价关系；此外，Peigen能生成新的S-boxes，满足给定的与安全有关和/或与性能有关的标准。在此，我们总结了Peigen的优势和局限。

Peigen在以下三个方面具有非常丰富的功能：

1. **评估**：给定一个n-bit S盒(3≤n≤8, 以下如不特殊说明则都满足该条件)的集合，Peigen评估了它们大多数与安全有关的属性（例如：差分分布表（Differential Distribution Table, DDT）、回旋镖相关表（Boomerang Connectivity Table, BCT）、线性逼近表（Linear Approximation Table, LAT）、代数正规表达式（Algebraic Normal Form , ANF）、自相关表（Auto-Correlation Table, ACT）、线性结构（Linear Structure, LS）、(v，w) -线性度((v, w)-linearity),，以及许多与这些表相关的详细标准。给定n位S-boxes，Peigen评估了它们的等价关系，包括Permutation-XOR等价（Permutation-XOR equivalence, PXE），线性等价（linear Equivalence, LE），Affine等价（Affine Equivalence , AE）。此外，对于一个给定的4位S-box，它可以将其AE类划分为PXE类。
2. **实现**：给定一个n-bit S盒的集合以及特定的实现方法的配置，PEIGEN可以生成在以下方面具有优越性的实现方法，包括BGC(Bitslice Gate Complexity) 、GEC (Gate Equivalent Complexity )、Gate Equivalent Complexity (GEC), Multiplicative Complexity (MC), and Depth Complexity (Depth)。
3. **生成**：给定给出一组标准和一组S盒，Peigen筛选出符合这组标准的好的S-boxes；如果给定一组标准，Peigen生成符合这组标准的新的S-boxes（包括与安全有关的和与实施有关的属性）。

除了众多的功能外，Peigen的开发还考虑到了效率、可扩展性和兼容性。它在评估安全属性时非常有效。

甚至对于3≤n≤8的n位S-boxes的大集合也是如此。在寻找实现和生成新的n位S-boxes时，它比现有的n=3、4的工具更有效率。虽然，它只支持3≤n≤8位的S-boxes，这主要是由于应用了特定的优化技巧，但它可以扩展到支持更大的n位数。

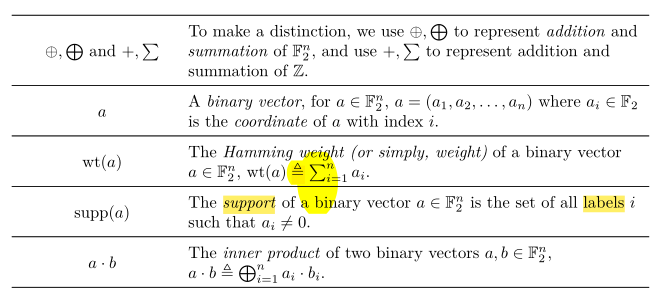
Peigen的主要限制是，它支持但往往是不可行的，除了非常轻量级的5位S-boxes，要找到5≤n≤8的实现和生成n位S-boxes。

利用该平台，我们对一些安全和性能优点之间的相互联系进行了一些有趣的观察。除了开发平台，我们还提供了比以前的工具更好的算法，这使得我们能够找到比以前的工具更好的S-boxes或更好的S-box实现。

**路线图**：本文的其余部分组织如下。第2节给出了本文中所使用的必要的前言和符号。第3节列出了所有可能的设计标准，包括安全和性能考虑。第4节介绍了我们的工具Peigen，然后是第5节的一些评估结果。最后，第6节讨论了一些未解决的问题并总结了本文。一些符号将被推迟到附录中。

2、符号说明

我们列出本节中使用的一些符号，如下。完整的符号清单可以在附录A中找到。



布尔函数是非常常见和有用的数学工具，用于设计和分析对称密钥加密原件。在布尔函数上测量的密码学标准与在对称密钥基元上测量的标准密切相关，特别是对于S-boxes。因此，在下面的章节中，为了介绍S-boxes的标准，我们通常首先提供一些布尔函数的定义和标准测量。

**布尔函数**： 一个n个二进制变量的布尔函数



它可以用几种方式表示。

直接地，我们可以使用真值表。我们有时用f直接表示它的值。当我们使用输入的顺序时，它是对应于f的所有值的向量。由于有2n个输入如果n变量布尔函数f的值向量wt(f)的汉明权重等于2n-1，即输出是均匀分布的，则称其为平衡的。平衡性通常是对用于加密基元的布尔函数的一个基本要求。

在数学上，我们可以使用多变量多项式。表示布尔函数的一个常规形式是代数正规式。