周报

2026-01-12

本周研究摘要

硕士学位论文第一章绪论与第二章基础知识的撰写工作

第二章完成了 SPN 结构密码基本原理、密码算法的软件实现技术和硬件实现技术三个方面的基础知识论述

第一章绪论撰写

该章节包括选题背景及研究意义、国内外研究现状和研究内容三个主要部分

- 选题背景：SPN 结构密码在现代分组密码设计中的主流地位及物联网安全领域应用需求
- 国内外研究现状：轻量级密码硬件实现、比特切片密码软件实现、GPU 并行密码实现
- 研究内容：三个主要研究方向的技术创新点

第二章基础知识撰写

系统介绍了 SPN 密码基本原理、软件实现技术和硬件实现技术三个方面的理论基础

- SPN 密码基本原理：S 盒变换、线性变换、AES-128 四个基本操作
- 软件实现技术：查找表实现、比特切片实现、SIMD 向量化实现
- 硬件实现技术：ASIC/FPGA 平台、迭代/串行/展开架构

比特切片技术的核心思想是将密码操作映射到位级布尔运算

第一章绪论

第一章 绪论

1.1 选题背景及研究意义

这是你的**论文**的介绍部分。

1.2 国内外研究现状

SPN 结构密码的高效实现研究涵盖硬件实现、软件实现和并行计算三个主要方向，国内外学者在各方向均取得了丰富的研究成果。

1.2.1 轻量级密码硬件实现研究现状

近年来，轻量级密码学领域受到广泛关注，如 PRESENT[2]、LED[3]、Midori[4]、QTL[5]、GIFT[6]、CRAFT[7]、Shadow[8]、DULBC[9]、IVLBC[10]、BipBip[11]和 LELBC[12]等算法相继被提出，更多算法可参考文献[13]。物联网安全挑战在文献[14]中有详细讨论，相关密码学技术见文献[15]，轻量级密码学综述见[16]。

在硬件实现方面，已有学者针对不同密码提出了优化架构。Lara 等[17]提出了 PRESENT 的 16 位数据通路架构，Pandey 等[18]优化了密钥调度，Shahbazi 等[19]提出了 AES 的 8 位串行架构，Li 等[20]针对 PRINCE 提出了展开与低成本架构，Bharathi 等[21]扩展了 PRESENT 的密钥长度，Yang 等[22]通过共享组件优化了 LILLIPUT。

在旁路攻击防护方面，差分故障分析（DFA）最早由 Biham 等[23]提出，后续在文献[24]中有更详细阐述。为应对故障攻击，Kaur 等[25]提出了错误检测机制，Canto 等[26]在后量子密码领域提出了专用检测方法。

1.2.2 比特切片密码软件实现研究现状

比特切片技术最初由 Biham[23]引入，作为在 64 位复杂指令集计算机上实现 DES 密码的高效方法。针对旁路攻击的安全增强措施已被广泛应用于密码算法的加固[27-28]。文献[29]开发了用于内存加密的优化 AES 实现。

比特切片实现在软件环境中已获得广泛应用[30]。通过比特切片技术优化基于 SPN 的密码实现，在利用单指令多数据（SIMD）指令的高性能 CPU 平台上取得了重要进展[31-33]。在图形处理器（GPU）环境中，已开发出高效的 AES-CTR 和 AES-ECB 实现[34]。

针对 32 位处理器的优化工作主要集中在线性层和非线性层两个方面。在线性层方面，Leurent 等[35]提出的启发式搜索方法将复杂的矩阵操作分解为更简单的异或和旋转任务。在非线性层方面，门复杂度最小化技术减少了 S 盒计算所需的逻辑操作数量[36]，布尔可满足性求解器被用于优化 S 盒变换[37-38]。32 位处理器的架构约束需要量身定制的优化策略[39-40]。

以确保每个输入比特的变化能够影响多个输出比特，从而实现扩散特性。这种比特扩散在多轮迭代后形成雪崩效应（Avalanche Effect），即明文的单个比特变化最终会影响密文的约一半比特。

对于大规模参数（$n \geq 128$），线性变换通常被分解为较小的子矩阵，如 4×4 或 8×8 的循环矩阵，以降低实现复杂度并提高计算效率。这种分解策略在保证扩散特性的同时，能够利用循环矩阵的特殊结构简化硬件实现和软件优化。

先进加密标准（Advanced Encryption Standard, AES）[41]是目前应用最广泛的 SPN 结构密码，已成为事实上的分组密码国际标准。AES 支持 128 比特的分组大小和 128、192、256 比特的密钥长度，分别对应 10 轮、12 轮和 14 轮的迭代加密。本节以 AES-128 为例，详细分析其 SPN 结构的设计原理和实现方式。

AES-128 将 128 比特的明文组织为 4×4 的状态矩阵（State），每个矩阵元素为 1 个字节（8 比特）。状态矩阵按列优先顺序排列，即第一列包含明文的前 4 个字节，第二列包含接下来的 4 个字节，依此类推。每一轮对状态矩阵进行四个连续的操作，实现混淆和扩散。AES-128 的单轮函数由以下四个操作组成：

- **SubBytes（字节替代）**：对状态矩阵的每个字节独立地应用 8 比特 S 盒，实现非线性替代。该操作相当于 16 个并行执行的 S 盒变换，提供了混淆特性。
- **ShiftRows（行移位）**：对状态矩阵的行进行循环左移。第 0 行不移位，第 1 行左移 1 字节，第 2 行左移 2 字节，第 3 行左移 3 字节。该操作提供跨列的扩散，使得每列的字节分散到不同的列中。
- **MixColumns（列混合）**：对状态矩阵的每一列独立地进行有限域 $GF(2^8)$ 上的矩阵乘法。该操作通过一个 4×4 的 MDS（最大距离可分）矩阵实现列内的扩散，确保每列中单个字节的变化影响该列的所有 4 个字节。
- **AddRoundKey（轮密钥加）**：将当前轮的轮密钥与状态矩阵进行按比特异或运算，将密钥信息混入状态。轮密钥通过密钥编排算法从主密钥派生。

如图 2-2所示，SubBytes 操作通过 S 盒的非线性特性实现混淆，使得攻击者难以建立密钥与密文之间的线性关系。ShiftRows 和 MixColumns 的组合实现扩散特性，确保单个字节的变化能够迅速影响整个状态矩阵。ShiftRows 提供跨列扩散，使得每行的字节分散到不同的列；MixColumns 提供列内扩散，使得每列中的字节充分混合。两者结合，使得任何输入字节的变化在两轮后能够影响整个状态矩阵。

AES-128 通过 10 轮迭代实现所需的安全强度。单轮 AES 变换不足以抵抗差分密码分析和线性密码分析等攻击，但 10 轮迭代使得任何微小的输入变化都能在输出中产生雪崩效应，满足 Shannon 的混淆和扩散原理。AES 自 2001 年成为标准以来，经历了广泛的密码分析测试，至今未发现有效的实用攻击方法，证明了其设计的安全性和可靠性。

第二章基础知识

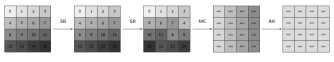

图 2-2: AES-128 单轮结构

SPN 结构的安全性来源于替代层和线性层的协同作用以及多轮迭代的累积效应。单轮 SPN 变换提供有限的混淆和扩散，不足以抵抗现代密码分析攻击。多轮迭代能够指数级提升密码强度，使得攻击者需要分析的状态空间呈指数增长，从而确保实用安全性。

S 盒提供的混淆特性使得攻击者难以建立密钥与密文的线性或仿射关系，而线性层的扩散特性确保了明文的微小变化能够在多轮后影响整个密文。这种替代与置换的交替结构，正是 Shannon 混淆和扩散原理的数学实现[43]。现代 SPN 密码如 AES 通过精心设计的 S 盒和线性层，使得差分密码分析和线性密码分析的复杂度远超穷举攻击，达到了实用安全性的要求。

SPN 密码的设计需要在 S 盒大小、轮数、实现开销和安全性之间进行权衡。较小的 S 盒（如 4 比特）实现开销低但需要更多轮数以达到相同的安全强度，而较大的 S 盒（如 8 比特）单轮安全性更高但实现复杂度增加[6,41]。设计者需要根据应用场景的具体需求，在这些因素之间寻求最优平衡。

2.2 密码算法的软件实现技术

密码算法的软件实现是指将密码算法映射到通用处理器上执行的过程。与专用硬件实现相比，软件实现具有灵活性强、开发周期短、易于更新维护等优势，广泛应用于服务器、个人计算机、智能手机及物联网设备等平台[44]。然而，软件实现通常面临处理器资源有限、指令集约束等挑战，需要针对具体平台进行优化以获得最佳性能。本节介绍密码算法软件实现的主要技术，包括查找表实现、比特切片实现和 SIMD 向量化实现等。

查找表（Look-Up Table, LUT）是密码算法软件实现中最直观的方法，通过预先计算并存储密码变换的输入输出映射关系，将运行时的计算转换为内存访问操作。对于 S 盒等非线性变换，查找表实现能够以 $O(1)$ 的时间复杂度完成任意输入到输出的映射。给定变换 $T : \mathbb{F}_2^m \to \mathbb{F}_2^n$，其查找表 LUT_T 定义为一个包含 2^m 个元素的数组，满足 $\mathrm{LUT}_T[x] = T(x)$，其中 x 为 n 比特输入。对于 AES 等分组密码，查找表技术可进一步扩展为 T 表（T-Table）实现，将 S 盒变换与列混合操作合并为四个预计算的查找表，每个表包含 256 个 32 比特条目，使得 AES 的单轮加密仅需 16 次查表操作和若干异或运算[41]。查找表实现的主要优势在于实现简单、执行速度快，但存在内存占用较大、缓存时序侧信道泄漏风险等局限性[45]。

比特切片（Bit-slicing）技术最初由 Biham[23]提出，用于在 64 位 CISC 处理器上高效实现 DES 密码。该技术的核心思想是将密码操作映射到位级布尔运算，利用处理器的位并行

非门数量。针对轻量级密码中常用的 4 比特 S 盒，研究者提出了多种低面积实现方案。通过 SAT 求解器结合 GEC 编码方案，可以找到面积最优的 S 盒实现。例如，对于 CRAFT 密码的 4 比特 S 盒，采用 MAOI1 和 MOAI1 等复合门可以有效减少门数量，实现面积优于传统方法[51]。

线性层负责实现扩散特性，其硬件实现的优化对整体电路性能有重要影响。比特置换（Bit Permutation）是最简单的线性变换，仅重新排列输入比特的位置而不改变比特值，在硬件实现中可以通过连线实现，不需要任何逻辑门资源。例如，PRESENT 密码的置换层可以通过纯连线实现，面积开销为零。然而，比特置换的"免费"实现仅在全宽度数据通路架构中成立，在串行架构中，由于数据通路宽度小于分组宽度，比特置换需要通过移位寄存器或多路选择器实现，会引入额外的面积和延迟开销。复杂的线性变换如 AES 的 MixColumns 操作需要通过矩阵乘法实现。在有限域 $\mathbb{F}_2$ 上，矩阵乘法可以分解为异或运算的组合，优化矩阵乘法实现的关键在于减少异或门的数量。对于具有特殊结构的矩阵（如循环矩阵、对合矩阵），可以利用矩阵的代数性质简化实现。例如，CRAFT 密码的 MixColumns 矩阵 M 是对合矩阵（即 $M^2 = I$），加密和解密可以共用同一电路，简化了实现复杂度。在串行架构中，MixColumns 操作需要特殊处理，由于每次只处理部分数据，需要使用寄存器暂存中间结果，并通过有限状态机控制数据流。

硬件实现中的控制单元负责协调各组件的工作时序，生成选择信号和使能信号，通常采用有限状态机（Finite State Machine, FSM）实现，状态数量取决于加密流程的复杂度。时钟门控（Clock Gating）是降低动态功耗的有效技术，通过在时钟信号路径上插入门控逻辑，在组件空闲时关闭其时钟信号，可以消除不必要的信号翻转，从而降低动态功耗[13]。在密码硬件实现中，可以对状态寄存器、密钥寄存器和中间计算单元分别应用时钟门控，在不同的加密阶段选择性地使能相应组件。

密码算法的硬件实现除了需要满足功能正确性外，还需要考虑抵抗侧信道攻击的能力。侧信道攻击通过分析密码设备在运行时泄露的物理信息（如功耗、电磁辐射、运行时间等）来推断密钥信息。常见的硬件层面防护措施包括掩码技术（Masking）、隐藏技术（Hiding）和冗余设计等。掩码技术将敏感中间值与随机掩码进行异或，使得泄露的物理信息与密钥无关；隐藏技术通过均衡功耗或添加噪声来掩盖与密钥相关的功耗特征；冗余设计通过硬件冗余实现错误检测，防止故障注入攻击。CRAFT 密码在设计时即考虑了抵抗差分故障攻击的能力，其线性层的特殊结构使得故障传播具有可预测性，便于实现高效的错误检测机制[7]。在实际部署中，需要根据安全需求和资源约束选择合适的防护方案。

2.4 本章小结

总结

下周计划

- 继续撰写大论文
- 重新梳理第四篇小论文

老师评语

小论文没有进展报告了，做事要让别人放心，不是自私让自己方便

后续每周汇报小论文进展