****

**硕士学位论文开题报告**

姓 名： 向嘉豪

学科（专业）： 电子信息

研 究 方 向： 密码工程学

选 题 名 称： 轻量级分组密码的软

硬件优化研究与实现

指 导 教 师: 李浪 教授

填表时间 2024年 月 日

|  |
| --- |
| **一、论文选题的目的和意义**  **1.选题来源**  随着基础通信设施的不断完善，如5G网络的普及，人们对通信的需求逐渐增加。这种需求的增长催生了大量基于通信的应用，如工业互联网、车联网、物联网等。然而，随着这些网络中传输的数据量的增加，安全问题变得越来越严重。这主要是由网络数据中数据包传输机制引起的。幸运的是，对称加密算法提供了一种解决方案，只需双方协商好密钥，就能确保数据在公共信道下的安全传输，而无需调整传输机制。  对称加密算法中，最广泛使用的是2001年的AES国际算法。它在WEB、WIFI等领域，以及服务器与个人计算机上都得到了广泛的应用。然而，为了避免AES算法存在的未知门陷，我国在2006年提出了SM4对称加密算法来替代AES算法。值得一提的是，SM4在2021年正式成为ISO标准并得到了国际认可。这些传统的对称算法在计算资源充足的场景下，能够提供较高的安全性。但在计算资源受限的场景下，由于这些算法的计算复杂度较高，因此需要一种更加轻量级的对称加密算法来满足这种场景的需求。  美国国家标准局NIST于2019年启动了轻量级密码算法LWC的征召。在56个轻量级密码算法的竞争中，ASCON [1] 算法经过三轮筛选，最终脱颖而出。到了2023年，ASCON算法已经成为NIST的轻量级加密标准。由于ASCON算法在资源受限的场景下表现出较高的性能，因此在物联网、车联网等场景中具有广泛的应用前景。  相较而言，我国在轻量级加密算法上的起步较晚，暂时还没有自己的轻量级加密算法标准。然而，在2019年由中国密码学会组织的全国密码算法设计竞赛中，一等奖获得者是一种名为uBlock [2]的轻量级加密算法。这种算法在计算资源充足的环境下表现优秀，同时在资源受限的场景下，也展现出了高性能。尽管如此，uBlock算法并未像ASCON算法那样得到广泛的学者关注，这使得我国在轻量级加密算法上与国际顶尖水平存在较大的差距。  这种差距不仅体现在轻量级加密算法的设计上，也同样存在于其实现上。这包括硬件和软件实现。具体来说，需要考虑如何高效设计硬件加密的IP核，并将其集成入芯片，以实现硬件级别的安全。同时，也需要考虑如何将加密算法高效实现于8-bit或32-bit的微控制器中，以确保应用的软件级别安全。我国在这个领域的研究相对较少。 |
| **2.文献回顾或综述**  在上世纪的算法设计中，对称加密算法的设计主要考虑了安全性。然而，这些设计往往较少考虑其在资源受限的场景下的性能。例如，DES [3]的实现部分提及其在软件与硬件的实现，但并未给出具体的参考实现方案与实现性能。本世纪网络数据传输量的增加使得网络安全问题日益严重。对称加密算法是一种能够确保数据在公共信道下安全传输的解决方案。然而，传统的对称加密算法在计算资源受限的场景下计算复杂度较高。因此，需要一种轻量级的对称加密算法来满足这种场景的需求。  在2011年之前，轻量级加密算法的研究主要集中在硬件实现上。这是因为硬件实现能够提供更高的性能，如ISO的轻量级加密标准PRESENT [4]。同时，硬件实现也能确保算法的安全性。然而，随着应用场景的增加，性能需求也在变化。现在，不仅关注硬件的电路面积，也开始关注软件加密的执行时间，如美国国家安全局（NSA）2013年提出的SIMON [5]。此外，还关注硬件加密的功耗，如2015年亚密会提出的Midori [6]，以及硬件加密的时延，如2016年美密会提出的SKINNY [7]。因此，轻量级加密算法的内涵正在不断扩展。  轻量级加密算法的设计和实现都在不断发展和完善。早期的轻量级加密算法主要集中在硬件实现上。例如，Arich等人对DES算法的硬件实现 [8]，他们将DES的加密速率提高到了1 Gbits/s。随着硬件实现的进步，性能指标也从电路面积和加密吞吐量扩展到了功耗和时延等方面。为了实现这些性能指标，提出了一些全新的硬件实现技术。其中，串行、展开、迭代、流水线等技术得到了广泛应用。  串行实现的核心思想是降低加密时的数据带宽。这种方法可以实现对加密组件的重复使用，并减少电路中的寄存器使用数量，以实现更低的电路面积。Priya等人提出了AES算法的串行实现 [9]。他们在Xilinx Spartan-II FPGA （XC2S15）上实现了面积资源174 slices和两个内存块。在这种实现中，最小的数据带宽为一个字节。为了进一步降低数据带宽，Leong等人提供了IDEA算法的比特级数据带宽的串行实现 [10] 。然而，对于以S盒作为基本加密组件的加密算法，实现一比特级的数据带宽是困难的。为了解决这个问题，Jean等人为类SPN结构的加密算法提出了比特滑动技术 [11] 。这种技术可以实现比特级的数据带宽。  展开实现的核心思想是在同一周期内计算加密算法的多轮函数。这种方法可以降低加密所需的延迟，并在引入更多面积的情况下获取更高的吞吐量。Zodpe等人在AES算法竞选的最终轮中使用了展开实现 [12]。他们的2轮展开实现Rijndael算法实现了最高的吞吐量，这在对比基于轮的迭代实现时尤为明显。Gupta等人将展开实现与流水线结合 [13] ，这极大地提高了RC4 [14] 算法的吞吐量。  迭代实现的核心思想是在同一周期内运算加密算法的一轮函数。通过反馈机制，可以实现多轮函数的计算。这种方法可以减少加密所需的面积，但会增加加密所需的延迟。值得注意的是，迭代实现在面积与吞吐量之间实现了较好的平衡。在AES算法竞选的最终轮中, Zodpe等人的迭代实现的Rijndael获得了最高的吞吐量与面积比 。类似的情况也出现在LWC竞赛中。ASCON算法的迭代实现在面积与吞吐量上取得了较好的平衡。  流水线实现的核心思想是将加密算法的一轮函数分解为多个阶段。这样，每个阶段的计算可以并行进行，实现多个分组同时加密。这种方法可以显著提高加密所需的吞吐量，但会增加实现的面积。Kryjak等人运用了这种流水线技术 [15]。他们对CLEFIA算法[16]进行了实现，极大地提高了加密的吞吐量。然而，考虑到所需的资源，这种实现主要适用于超性能计算场景。  轻量级加密算法的软件实现主要集中在微控制器上。这与硬件实现有所不同。微控制器提供了更高的灵活性。例如，可以在不同的应用场景下调整加密算法的参数，或者使用不同的加密算法。同时，微控制器也提供了更高的可移植性。例如，可以在不同的硬件平台上运行加密算法。然而，相比硬件实现，软件实现的性能指标通常较低。  早期的轻量级加密算法的软件实现主要集中在8-bit微控制器上。这些微控制器常用于RFID和传感器等设备。在1991年，Merkle尝试将Khufu算法进行软件实现。这为算法在其他不同场景下的应用打开了新的可能性 [17] 。后来，Osvik将AES算法实现在8-bit的AVR微控制器上，刷新了当时的最快加密记录 [18] 。  随着微控制器的发展，32-bit微控制器成为主流。软件实现的侧重也发生了改变。Rogaway开始在32-bit的现代微控制器上，对SEAL算法进行了软件实现 [19]。Bertoni首次将AES算法实现在32-bit的微控制器上，并通过多种架构的仿真器对优化实现进行验证 [20]。Schwabe在结合ARM平台在Cortex-M3和M4上实现了AES算法，扩展了AES算法的用途[21]。  在软件实现中，存在一些特殊的情况需要注意。例如，抵抗侧信道攻击通常需要添加额外的操作。如何更高效地实现这些抗攻击的操作，成为了一个重要的研究方向。例如，Rivain提出了一种高阶掩码技术 [22]。这种技术可以有效地抵抗设备在侧信道攻击中的相关功耗攻击。另一方面，轻量级加密算法在某些场景下具有优势。例如，在区块链等对加密算法吞吐量要求较高的场景下，轻量级加密算法可以发挥重要作用。利用SIMD指令集可以提高加密算法的吞吐量 [23] 。此外，也可以利用GPU来实现加密。这些都是软件实现的重要方向。  相比硬件实现和软件实现，软硬件协同实现的研究相对较少。这主要是因为软硬件协同实现需要同时考虑硬件和软件的特性。然而，软硬件协同实现在保证算法性能的同时，也能提高算法的灵活性。例如，当加密算法的标准发生变化时，软硬件协同实现能够更快地适应这些变化。值得注意的是，与硬件实现和软件实现相比，软硬件协同实现起步较晚。  在早期的实现中，加密算法的硬件实现和软件实现通常是分开的。例如，加密算法作为IP核心，以片上外设的形式挂载在数据总线上，如Usselmann实现的AES算法。由于外设与CPU之间的通信开销较大，这种实现方式的性能较低。为了提高性能，一些研究开始尝试将加密算法的硬件实现与软件实现集成在一起。这种实现方式可以减少外设与CPU之间的通信开销，从而提高性能。然而，由于各家厂商的CPU架构不同且闭源，这种实现方式的研究较少。  开源硬件的发展为软硬件协同实现的研究带来了新的机遇。例如，RISC-V是一种开源的CPU架构，可以自由使用。这种架构的出现，为软硬件协同实现提供了新的研究方向。Marshall设计了一套轻量级扩展指令集，将ChaCha算法实现在RISC-V上[24]。基于扩展指令集，Chen在2023年提出了一组通用扩展指令集在RISC-V之上，并将其运用到了LWC最终轮的10个算法上[25]。这种方式将加密的硬件电路集成入了CPU的执行流水线中，保证了加密操作的速度。同时，加密算法中的通用操作被设计为对应的指令，这样即使加密标准更新，也能保持对加密算法的加速效果。 |

|  |
| --- |
| **3.研究视角和意义**  **1、研究视角**  **（1）轻量级加密算法硬件实现**  对于传统的性能指标，如面积和吞吐量，轻量级加密算法的硬件实现已经有了较好的解决方案。然而，对于新的性能指标，如功耗和时延，还需要进一步的研究。未来的研究重点是如何提出新的硬件实现技术，以适应这些新的性能指标。  **（2）轻量级加密算法软件实现**  软件实现的技术比硬件实现更为丰富。因此，许多研究都集中在如何将加密算法高效地实现在更先进的平台上。未来的研究重点将是开发新的软件实现技术，以适应这些新的平台。同时，也需要考虑如何预防新的攻击方式。  **（3）轻量级加密算法软硬件协同实现**  软硬件协同实现为轻量级加密算法的实现开辟了新的研究方向。这种实现方式将不利于软件实现的部分转化为硬件实现，并以此作为基础操作。然后，利用软件指令来构建整体算法，从而在加密算法的性能与灵活性之间取得平衡。这也为算法设计提供了一条全新的思路。未来的研究重点是将轻量级加密算法中的通用操作设计为扩展指令集或通用加密IP核。  **2、研究意义**  **（1）理论意义**  本研究旨在将密码组件转化为高效运行的软硬件实现，构建密码组件在软硬件单元上的理论模型，基于优化理论提高轻量级加密算法，在不同场景下的理论性能，为未来研究者提供可参考的优化思路与理论依据。  **（2）方法意义**  研究轻量级加密算法在专用集成电路（ASCI）和现场可编程门阵列（FPGA）上的硬件实现。这将提高算法实现的性能，同时确保其硬件级别的安全。在8-bit或32-bit的微控制器上的软件实现。这将提高算法实现的性能，同时确保其软件级别的安全。软硬件协同实现。在确保算法实现的灵活性的前提下，最大程度地提高算法实现的性能。  **（3）应用意义**  硬件实现主要适用于专用集成电路（ASIC）和现场可编程门阵列 （FPGA）。在对加密性能和成本有较高要求的场景下，硬件实现具有不可替代的地位。例如，可以应用于IC无线射频卡、云加密机等。软件实现主要适用于8-bit或32-bit的微控制器。在对加密算法部署灵活性有较高要求的场景下，软件实现具有广泛的应用前景。例如，可以应用于智能家居、车联网等。软硬件协同实现主要适用于同时对加密性能、成本和灵活性有较高要求的场景。例如，可以应用于区块链等。  **（4）社会意义**  对于不同的应用场景，可以提供各种实现方案，以实现更低的成本、更高的性能和更高的灵活性。此外，研究成果将推动我国在轻量级加密算法领域的研究进展，促进我国在轻量级加密算法实现上的独立自主。 |

|  |  |
| --- | --- |
| **4.研究的主要内容和重点**  **1、研究的主要内容**  本研究的内容是研究轻量级分组密码算法的多场景优化实现。研究轻量级分组密码在硬件、软件和软硬件平台下优化实现的关键技术，具体来说，包括以下三个方面的内容：  （1） 研究轻量级分组密码算法在专用集成电路（ASIC）和现场可编程门阵列（FPGA）上的硬件实现。  在需要高性能的场景下，硬件实现是不可替代的。其中就加密算法部署的规模，ASIC是一种专用集成电路，可以提供更高的性能，在大批量部署下具有优势。同时FPGA是一种现场可编程门阵列，在少批量部署上，可以提供更高的灵活性。本项目将研究轻量级分组密码算法在ASIC和FPGA上的硬件实现，以提高算法实现的性能，同时考虑硬件实现过程中泄露的侧信道信息，防止功耗、故障攻击等，为算法提供硬件实现安全。  （2）研究轻量级分组密码算法在微控制器上的软件实现。  在物联网、智能家居等场景下，终端设备采用资源受限的微控制器，将硬件实现的加密算法部署到微控制器上，需要片外加密芯片或片内加密模块，这样会增加成本。因此，软件实现是一种更为经济的解决方案。本项目将研究轻量级分组密码算法在8-bit或32-bit的微控制器上的软件实现，以提高算法实现的性能，同时考虑软件实现过程中泄露的侧信道信息，防止时间、缓存攻击等，为算法提供软件实现安全。  （3）研究轻量级分组密码算法的软硬件协同实现。  在一些场景下，需要兼顾硬件实现的性能和软件实现的灵活性。例如，区块链等场景下，需要高性能的加密算法，同时也需要灵活的加密算法。本项目将研究轻量级分组密码算法的软硬件协同实现，将加密算法的通用操作设计为扩展指令集或通用加密IP核，以提高算法的性能和灵活性。  **2、研究的重点**  （1）研究轻量级分组密码组件与结构对于硬件实现的影响。在硬件实现原语之上，优化实现轻量级分组密码，同时设计硬件实现的侧信道防护方案。  （2）研究轻量级分组密码算法微控制器上的软件实现。在微控器的基础指令集与扩展指令集上，优化实现轻量级分组密码，同时设计软件实现的侧信道防护方案。  （3）研究轻量级分组密码算法的软硬件协同实现。利用硬件优化技术设计高效扩展指令集或通用加密IP核，结合软件优化技术提高算法部署的性能和灵活性，同时给出实现的侧信道防护方案。 | |
| **二、研究方案**  **1.研究方法、技术方案（技术路线，技术措施）**  **1. 研究方法**  （1） 研究轻量级分组密码算法的硬件实现。  在硬件实现方面，本项目将研究轻量级分组密码算法的硬件实现技术，从算法架构、组件优化和安全实现出发，确保算法实现的性能与硬件安全。在算法架构层，结合轻量级分组密码的特点，优化实现轻量级分组密码。举例来说，将CRAFT算法进行串行架构实现，分析调整加密组件的执行次序，同时对加密组件进行时序状态分析，确保算法高效运行，最终减少数据带宽，降低硬件实现面积。在组件优化层，对轻量级分组密码的S盒、P盒等组件进行优化，提高组件的性能。举例来说，将CRAFT算法的S盒在布尔满足性（SAT）上进行建模，约束S盒中的电路门数，求解出低面积S盒硬件实现。在硬件安全上，对于故障攻击，通过冗余检测电路和预计算编码表确保安全，在功耗攻击上，通过随机运算电路和掩码技术确保安全。  （2） 研究轻量级分组密码算法在微控制器上的软件实现。  在软件实现方面，从软件运行的平台、加密算法组件优化和安全实现出发，确保算法实现的性能与软件安全。在软件运行的平台层，本项目将研究轻量级分组密码算法在主流32-bit的微控制器上的实现，包括ARM-Cortex M3、M4、RISC-V等。结合精简指令集平台所具备的基本指令，对算法进行汇编优化，同时探究特殊平台下的指令对算法的影响，如ARM-Cortex M4中循环移位指令对算法的加速效果。在加密算法组件优化层，对轻量级分组密码的S盒、P盒和扩散矩阵等组件进行优化，提高组件的性能。举例来说，将CRAFT算法的S盒在可满足性（SMT）上比特切片建模，通过限制模型的指令数，提高S盒的执行效率。在软件安全上，对于时间攻击，通过随机延迟确保安全，在缓存攻击上，通过比特切片减少算法内存使用，保证组件相同运行时间，减少侧信道信息泄露。  （3）研究轻量级分组密码算法的软硬件协同实现。  在软硬件协同实现上，本项目将研究轻量级分组密码算法的软硬件协同实现技术，在开源的RISC-V平台上设计高效扩展指令集及通用加密外设。在扩展指令集上，结合寄存器的数量限制以及指令的设计准则，将加密算法中的通用操作设计为扩展指令，例如，加密算法中起到扩散功能的线性层，设计比特级置换指令，对起到混淆功能的非线性层，设计可配置的替换指令。在通用加密外设上，可以在不更改芯片核心流水线的情况下，将通用加密硬件电路集成到芯片中，将加密算法的通用操作设计为外设访问，通过配置外设的配置寄存器，实现不同加密算法的加速。同时，设计软件接口，将扩展指令集和通用加密外设的操作封装为软件指令，提供给软件开发者使用。 | |
| **2.结构安排和主要内容**  第一章 绪论，包括课题研究背景意义，国内外研究现状，主要的研究与创新，以及文章的组织结构。  第二章 相关的理论背景。介绍软硬件实现轻量级算法的相关理论。  第三章 轻量级算法的硬件实现。  第四章 轻量级算法的软件实现。  第五章 轻量级算法的软硬件协同实现。  随后 参考文献、攻读学位期间取得的研究成果和致谢。 | |
| **3.实施方案所需的条件（技术条件，实验条件）**  实验主要所需要软件、硬件实验平台。包括：小型MCU微控制器、中端FPGA测试板等。 | |
| **4.预期能达到的目标；可能的创新；存在的主要问题和难点**  **1、预期能达到的目标**  发表SCI论文3篇   1. **可能的创新**   i）轻量级S盒在不同工艺库下的最优实现，ii）轻量级S盒最优比特切片实现，iii）通过硬件技术设计出通用加密组件。   1. **存在的主要问题和难点**   无 | |
| **三、研究计划进度**  2024.6-2025.6  完成实验创新点的书写以及实验。  2025.6-2025.12  完成大论文书写。  2025.12-2026.3  完成答辩。 | | |
| 1. **参考文献** 2. Dobraunig C, Eichlseder M, Mendel F, et al. Ascon v1. 2: Lightweight authenticated encryption and hashing[J]. Journal of Cryptology, 2021, 34: 1-42. 3. Wu W L, Zhang L, Zheng Y F, et al. The block cipher uBlock[J]. Journal of Cryptologic Research, 2019, 6(6): 690-703. 4. Alahdal A, Deshmukh N K. A systematic technical survey of lightweight cryptography on IoT environment[J]. International Journal of Scientific & Technology Research, 2020, 9(3): 1-30. 5. Sravya G, Kumar M O V P, Sheeba G M, et al. Hardware lightweight design of PRESENT block cipher[J]. Materials Today: Proceedings, 2020, 33: 4880-4886. 6. Rashidi B. High‐throughput and flexible ASIC implementations of SIMON and SPECK lightweight block ciphers[J]. International journal of circuit theory and applications, 2019, 47(8): 1254-1268. 7. Li Y, Wang M, Ou H, et al. Improved integral analysis on lightweight block cipher Midori[C]. 2019 IEEE 5th International Conference on Computer and Communications (ICCC). IEEE, 2019: 1494-1498. 8. Sevin A, Mohammed A A O. A survey on software implementation of lightweight block ciphers for IoT devices[J]. Journal of Ambient Intelligence and Humanized Computing, 2023, 14(3): 1801-1815. 9. Zeebaree S R. DES encryption and decryption algorithm implementation based on FPGA[J]. Indones. J. Electr. Eng. Comput, 2020, 18(2): 774-781. 10. Priya S S S, Karthigaikumar P, Teja N R. FPGA implementation of AES algorithm for high speed applications[J]. Analog Integrated Circuits and Signal Processing, 2022: 1-11. 11. Ledda M K C, Gerardo B D, Hernandez A A. Enhancing IDEA algorithm using circular shift and middle square method[C]. 2019 17th International Conference on ICT and Knowledge Engineering (ICT&KE). IEEE, 2019: 1-6. 12. Roldán Lombardía S, Balli F, Banik S. Six shades lighter: a bit-serial implementation of the AES family[J]. Journal of Cryptographic Engineering, 2021, 11(4): 417-439. 13. Zodpe H, Sapkal A. An efficient AES implementation using FPGA with enhanced security features[J]. Journal of King Saud University-Engineering Sciences, 2020, 32(2): 115-122. 14. Jiao L, Hao Y, Feng D. Stream cipher designs: a review[J]. Science China Information Sciences, 2020, 63(3): 131101. 15. Noura H, Chehab A. An efficient and secure variant of RC4 stream cipher scheme for emerging networks[C]. 2019 IEEE Wireless Communications and Networking Conference (WCNC). IEEE, 2019: 1-8. 16. Rashidi B. Efficient and flexible hardware structures of the 128 bit CLEFIA block cipher[J]. IET Computers & Digital Techniques, 2020, 14(2): 69-79. 17. Jangra M, Singh B. Performance analysis of CLEFIA and PRESENT lightweight block ciphers[J]. Journal of Discrete Mathematical Sciences and Cryptography, 2019, 22(8): 1489-1499. 18. Hendi A Y, Dwairi M O, Al-Qadi Z A, et al. A novel simple and highly secure method for data encryption-decryption[J]. International Journal of Communication Networks and Information Security, 2019, 11(1): 232-238. 19. Hajihassani O, Monfared S K, Khasteh S H, et al. Fast AES implementation: A high-throughput bitsliced approach[J]. IEEE Transactions on parallel and distributed systems, 2019, 30(10): 2211-2222. 20. van der Hagen M K, Lucia B. Client-optimized algorithms and acceleration for encrypted compute offloading[C]. Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, 2022: 683-696. 21. Dhanda S S, Jindal P, Singh B, et al. A compact and efficient AES-32GF for encryption in small IoT devices[J]. MethodsX, 2023, 11: 102491. 22. Kim H, Jang K, Song G, et al. SPEEDY on Cortex–M3: efficient software implementation of SPEEDY on ARM Cortex–M3[C]. International Conference on Information Security and Cryptology. Cham: Springer International Publishing, 2021: 434-444. 23. Ming J, Zhou Y, Li H, et al. A secure and highly efficient first-order masking scheme for AES linear operations[J]. Cybersecurity, 2021, 4: 1-15. 24. Xu R, Xiang Z, Lin D, et al. High-throughput block cipher implementations with SIMD[J]. Journal of Information Security and Applications, 2022, 70: 103333. 25. Marshall B, Page D, Pham T H. A lightweight ise for chacha on risc-v[C]. 2021 IEEE 32nd International Conference on Application-specific Systems, Architectures and Processors (ASAP). IEEE, 2021: 25-32. 26. Cheng H, Großschädl J, Marshall B, et al. RISC-V instruction set extensions for lightweight symmetric cryptography[J]. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2023: 193-237. | | |
| **五、审核意见：**    签名：  年 月 日 | | |
|  | | |