

**本科生毕业设计（论文）开题报告**

**学生姓名： 刘思宇**

**导师姓名、职称： 肖敏(副教授)**

**所属学院： 计算机与人工智能学院**

**专业班级： 计算机zy2001**

**设计（论文）题目： RSA加解密算法在FPGA虚拟仿真平台上的实现与应用**

2024年3月19日

|  |
| --- |
| **撰写内容要求（可加页）：**  1．目的及意义（含国内外的研究现状分析）  1.1设计目的及意义  在当今信息技术高速发展的时代背景下，数据安全和信息加密技术成为了网络安全的重要组成部分。随着互联网的普及和大数据时代的来临，人们越来越重视个人信息和数据的保护。RSA算法作为一种公钥加密技术，以其独特的加密机制和较高的安全性，在数字签名和加密通信等领域得到了广泛应用。然而，随着计算能力的提升和计算方法的进步，传统的软件实现方式在处理大规模数据加密时，面临着速度慢、资源消耗大等问题。因此，寻找一种更高效、更安全的RSA算法实现方式成为了研究的重点。  FPGA（现场可编程门阵列）技术，以其灵活的可编程性、较高的处理速度和较低的功耗，为解决上述问题提供了新的思路。将RSA算法在FPGA上实现，不仅可以利用FPGA并行处理的特点大幅提高加解密的速度，还可以通过硬件级别的实现增强算法的安全性。此外，FPGA的可重配置性使得加密系统能够根据不同的应用需求灵活调整，增强了系统的适应性和可扩展性。  本研究旨在探索RSA加解密算法在FPGA虚拟仿真平台上的实现与应用，通过设计和优化算法在FPGA上的硬件实现流程，既保证了加解密过程的高效率，又确保了数据传输过程的高安全性。同时，通过在虚拟仿真平台上的测试和验证，可以预先发现并解决实际应用中可能遇到的问题，为后续的实物部署和应用提供了坚实的基础。综上所述，本研究不仅对提升RSA算法在实际应用中的性能和安全性具有重要意义，也为未来FPGA在加密技术领域的应用提供了新的研究方向和广阔的发展前景。通过本研究的深入，我们期待为网络安全和信息保护领域提供更加高效、可靠的技术支持，为保护数字时代的信息安全作出贡献。  1.2国内外研究现状  1.2.1理论研究  Jan-Hendrik Evertse于1992年在JCRYPTOL发布了一篇论文[1],这篇论文指出,如果签名机构向个人发布某些类型的RSA签名, 在个人无法计算RSA根的假设下,根据不同类型RSA签名之间的内在联系,在某些条件下,可以从一些给定类型的签名推导出其他类型的签名.  Amos Fiat于1997年在JCRYPTOL发布了一篇论文[2],这篇论文提出了一种称为批量RSA(Batch RSA)的RSA算法变体,这项工作通过批量处理和分布式处理两种技术手段,有效改进了RSA算法的性能和安全性,这两个特性使得批量RSA成为一种非常有前景的RSA算法变体  Don Coppersmith于1999年在JCRYPTOL发布了一篇论文[3],这篇论文讨论了在中国剩余定理(CRT)算法的基础上实现公钥密码系统时,如果存在错误计算,可能导致的安全问题.,在某些特定情况下,仅需一个错误的签名,就足以恢复出密钥  Rosario Gennaro and Tal Rabin于2000年在JCRYPTOL发布了一篇论文[4],论文提出了一种在门限RSA签名中引入鲁棒性的新方法,可以容忍参与者的恶意行为,同时保持与原始方案相当的效率.  Rosario Gennaro and Tal Rabin于2000年在JCRYPTOL发布了一篇论文[5],这篇论文介绍了第一个基于RSA的不可否认签名方案  R. Fischlin and C. P. Schnorr于2000年在JCRYPTOL发布了一篇论文[6],这篇论文讨论了RSA加密函数的安全性证明  Daniel R. L. Brown于2016年在JCRYPTOL发布了一篇论文[7],这篇论文的主要结论是:如果因数分解是困难的,那么直线程序(straight line programs)不能有效地解决低公钥指数的RSA问题  Carmit Hazay于2017年在JCRYPTOL发布了一篇论文[8],这篇论文的主要贡献有两个:  提出了第一个在恶意攻击下安全的、完全可模拟的分布式RSA合数生成协议.之前的协议都是通用的,没有考虑RSA合数生成的特殊性.  在双方设置下,提出了一个完整的Paillier门限加密方案,可以抵御恶意攻击.之前的Paillier门限方案都是多方设置.  Atsushi Takayasu†于2019年在JCRYPTOL发布了一篇论文[9],这篇论文通过新颖的技术手段,在小CRT指数RSA的安全性分析领域取得了重要进展,  Megan Chen于2022年在JCRYPTOL发布了一篇论文[10],这篇论文提出了一个新的多方协议,用于分布式生成RSA的双素数模数,在假设遗忘传输和因数分解困难的情况下,可以抵抗任意子集的恶意合谋方.  1.2.2实际应用  早在1990年,J.R. Sherwood and V.A. Gallo在CRYPTO发表了一篇论文[11],这篇摘要讨论了在包含交互式和存储转发设施的复杂网络中,使用智能卡进行RSA数字签名的应用  在2008年,华中科技大学曹计昌团队在实验室预研项目"可动态重构的双内核多片内操作系统的智能卡体系结构"时,学生便提出了一种基于 FPGA 的 RSA 加密芯片的设计与实现[12]  2013年,大连理工大学王开宇团队将FPGA动态重构技术与RSA非对称加密技术结合,设计并实现了一个可利用以太网远程加密动态重配置的系统[13]  1.2.3改进实现  Thorsten Kleinjung于2010年在CRYPTO发布了一篇论文[14], 这篇论文主要报告了使用数域筛法(number field sieve)对768位数RSA-768进行因式分解的结果,并讨论了其对RSA的一些影响.  Ari Shawkat Tahir2015年在Xilinx\_Spartan3 (XC3S50器件, PG208封装, -4速度)上实现时[15],RSA算法设计仅占用29%的芯片资源,可实现的工作时钟频率为68.573 MHz  西安交通大学龚文杰团队为了满足某款低速微型芯片("墨盒特制芯片解决方案")的安全性需求,要设计一个小面积的RSA硬件加密模块,实现芯片数据保护,防止被破解.他们采用了基2免减蒙哥马利模乘算法[16],避免传统算法运算时间长、面积大的缺陷,实现了2048位RSA加密运算的硬件模块,吴克强团队[17]优化小素因子预筛选方法,提出新的预筛选处理方式,使每个小素数只需过滤一遍,提高预筛选速度.优化模逆运算,提出改进的Euclid算法,通过加减移位操作完成模逆,避免复杂除法运算,并通过有效处理提高运算速度,易于硬件实现.  Jianiong Tan等人2017年IEEE提出了一种基于Xilinx Zynq SoC异构架构的RSA加速方案[18],采用无减法Montgomery算法和中国剩余定理(CRT)实现高速RSA处理器,实验结果表明,总体性能与节点数量成线性关系.与传统的多核桌面(Intel i7-3770)相比,可实现6-9倍的加速;与多核服务器(288核)相比,性能相当.与传统平台相比,该方案的能效可提高2.5倍.  2．研究（设计）的基本内容、目标、拟采用的技术方案及措施  2.1基本内容  本研究的核心目的在于探索并实现RSA非对称加密算法在FPGA（现场可编程门阵列）平台上的加解密过程，并通过具体的应用案例展现其在现实世界中的应用价值和效益。为了确保研究的全面性和深入性，我们将分为以下三个主要部分进行研究：  理论学习： 在进行算法实现之前，首先需对密码学的基础知识进行全面的学习，特别是深入理解非对称加密算法中RSA算法的原理。RSA算法的核心在于利用大素数的选取和它们乘积的因式分解困难性，基于数论原理构建的公开密钥密码体制，使得公钥可公开用于加密，而只有持有私钥的用户才能解密。  算法实现： 在理论基础上，进一步在FPGA平台上实现RSA加密解密算法。FPGA的可编程性和高性能计算优势使其成为实现高效加密算法的理想选择。本部分的目标是通过FPGA平台的设计和编程，实现RSA算法的有效运行，并对其进行详细的仿真测试，验证算法的正确性和效率。  应用案例分析：  在成功实现RSA算法的FPGA仿真之后，将寻找并选择一个具体的应用场景，以此为例展示RSA加解密算法在FPGA平台上实现的实际价值。这一部分将详细说明所选应用案例的背景、需求以及RSA算法在此案例中的应用方式和优势。目标是展现通过FPGA平台实现RSA算法在现实世界安全通信、数据保护等领域的应用潜力和成果。  综上所述，本研究旨在通过FPGA平台的高效计算能力，实现并优化RSA加解密算法，从而提供一种既安全又高效的加密解密解决方案。通过理论学习、算法实现与应用案例的分析，本研究不仅加深了对RSA算法及其在硬件平台上实现的理解，也为密码学应用提供了新的视角和技术路径，有望在信息安全领域中发挥重要作用。  2.2技术方案  理论学习与原理掌握： 首先，深入学习RSA公开密钥密码体制的原理，包括大素数的选择、公钥与私钥的生成、加密与解密过程。理解RSA算法的数学基础是实现算法的前提。  FPGA平台选择与配置： 确定适合实现RSA算法的FPGA平台，包括硬件选择、开发环境搭建以及必要的软件配置。FPGA的可编程性和并行处理能力使其成为加密算法硬件实现的理想平台。  算法硬件化设计： 将RSA算法的运算过程转化为FPGA上的硬件逻辑设计，包括模数运算、加密和解密核心算法的硬件描述语言(HDL)编码，以及适配FPGA的资源优化。重点是将算法中的关键步骤如模幂运算高效实现。  仿真测试与验证： 在FPGA开发环境中进行仿真测试，验证算法实现的正确性和性能。这一步骤包括测试各种加解密场景，确保在不同的操作条件下算法都能稳定运行。  实际应用案例演示： 最后，选择一个或多个具体应用案例来展示RSA加解密算法在FPGA虚拟仿真平台上的实际运行效果和应用价值。案例的选择应体现算法实现的安全性、效率和可靠性。  3．进度安排  前期调研与准备（1月26日-3月10日）：  进行广泛的调研，查阅与RSA算法及FPGA实现相关的至少10篇文献，其中至少包含5篇英文文献。这一阶段的目标是深入理解RSA算法的原理和FPGA技术的应用，为项目的基本内容和技术方案的确定打下坚实的基础。完成开题报告，详细阐述研究目的、方法、预期目标及其科学和实用价值。  文献翻译与技术学习（3月10日-3月24日）：  翻译与项目相关的英文文献，确保翻译工作量不少于5000汉字。翻译工作完成后，提交给指导教师审阅并根据反馈进行修改。在这一阶段还需要开始深入学习FPGA设计和编程技术，为后续的算法实现打下技术基础。  RSA算法研究与FPGA开发环境搭建（3月25日-4月7日）：  深入研究RSA加解密算法的数学原理和实现方法，掌握关键技术点。选择合适的FPGA开发板和开发环境，完成开发环境的配置和基本操作训练。  算法设计与初步实现（4月8日-4月28日）：  设计RSA算法的FPGA实现方案，包括算法的硬件描述语言(HDL)编码和初步的功能模块设计。在FPGA开发环境中实现算法的初步版本，并进行基本的功能测试。  仿真测试与性能优化（4月29日-5月20日）：  对RSA算法的FPGA实现进行全面的仿真测试，识别并解决存在的问题。根据测试结果进行性能优化，提高算法的效率和稳定性。  论文撰写与提交（5月21日-6月5日）：  在指导教师的指导下，开始组织论文材料，撰写毕业设计论文初稿，要求字数为30000字以上。初稿完成后，根据指导老师的审查和建议进行修改，最终定稿。按照学校提供的论文规范要求，完成论文的最终撰写、打印和装订，准备提交。  通过这一系列细致的进度安排，项目旨在确保RSA加解密算法在FPGA平台的研究和实现不仅具有理论的深度和广度，而且能够在实践中展现出高效和实用的价值。此外，这样的安排有助于系统地推进研究工作，确保按时完成毕业设计的各个阶段任务。  4．阅读的参考文献不少于15篇（其中近五年外文文献不少于3篇）  [1] Evertse, J.-H., & van Heyst, E. (1992). Which New RSA-Signatures Can Be Computed from Certain Given RSA-Signatures? Journal of Cryptology, 41-52.  [2] Fiat, A. (1997). Batch RSA. Journal of Cryptology, 10(2), 75-88.  [3] Joye, M., Lenstra, A.K., & Quisquater, J.J. (1999). Chinese Remaindering Based Cryptosystems in the Presence of Faults. Journal of Cryptology, 12(4), 241-245.  [4] Gennaro, R., Rabin, T., Jarecki, S., & Krawczyk, H. (2000). Robust and Efficient Sharing of RSA Functions. Journal of Cryptology, 13(2), 273-300.  [5] Fischlin, R., & Schnorr, C. P. (2000). Stronger Security Proofs for RSA and Rabin Bits. Journal of Cryptology, 13(2), 221-244.  [6] Gennaro, R., Rabin, T., & Krawczyk, H. (2000). RSA-Based Undeniable Signatures. Journal of Cryptology, 13, 397-416.  [7] Brown, D. R. L. (2016). Breaking RSA May Be As Difficult As Factoring. Journal of Cryptology, 29(1), 220-241.  [8] Hazay, C., Mikkelsen, G. L., Rabin, T., Toft, T., & Nicolosi, A. A. (2019). Efficient RSA Key Generation and Threshold Paillier in the Two-Party Setting. Journal of Cryptology, 32(1), 265-323.  [9] Takayasu, A., Lu, Y., & Peng, L. (2019). Small CRT-Exponent RSA Revisited. Journal of Cryptology, 32(4), 1337-1382.  [10] Chen, M., Doerner, J., Kondi, Y., Lee, E., Rosefield, S., & Shelat, A. (2022). Multiparty Generation of an RSA Modulus. Journal of Cryptology, 35(12).  [11] Sherwood, J. R., & Gallo, V. A. (1990). The Application of Smart Cards for RSA Digital Signatures in a Network Comprising Both Interactive and Store-and-Forward Facilities. In S. Goldwasser (Ed.), Advances in Cryptology - CRYPTO'88 (pp. 484-496). Springer-Verlag Berlin Heidelberg.  [12] 段天. (2008). 基于FPGA的RSA加密芯片的设计与实现(硕士论文, 华中科技大学).  [13] 赵赟. (2013). RSA加密技术在FPGA动态重构系统安全方面的应用(硕士论文,大连理工大学).  [14] Kleinjung, T., Aoki, K., Franke, J., Lenstra, A. K., Thomé, E., Bos, J. W., Gaudry, P., Kruppa, A., Montgomery, P. L., Osvik, D. A., te Riele, H., Timofeev, A., & Zimmermann, P. (2010). Factorization of a 768-Bit RSA Modulus. In T. Rabin (Ed.), Advances in Cryptology - CRYPTO 2010 (pp. 333-350). Springer Berlin Heidelberg.  [15] Tahir, A. S. (2015). Design and Implementation of RSA Algorithm using FPGA. INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY, 16(8), 7141-7147.  [16] 龚文杰 (2016) , RSA算法的FPGA快速实现, (硕士论文,西安交通大学)  [17] 吴克强 (2017) , 面向硬件的RSA密钥快速生成方法设计及其FPGA实现(硕士论文,西安交通大学)  [18] Bai, X., Jiang, L., Dai, Q., Yang, J., & Tan, J. (2017). Acceleration of RSA Processes based on Hybrid ARM-FPGA Cluster. In 2017 IEEE Symposium on Computers and Communications (ISCC) (pp. 444-449). IEEE.  5．指导教师意见  指导教师（签名）：  年 月 日 |