2018 OpenCAPI+OpenPower异构计算设计大赛参赛指导及须知

**综述**

1. 同意并遵守本次大赛规则的高校计算机，微电子，信息技术等相关专业学生或者相关科研机构工作人员均可参赛；
2. 所有参赛者均需完整填写注册信息，填写不完整或填写错误、虚假报名将不被接纳；
3. 主办方保留进一步补充本规则的权利。如参赛者对此有异议，可选择退出大赛，并应及时通知主办者但无权要求主办者返还因参赛发生的任何费用。
4. 大赛主办方(排名不分先后)：国际商业机器(中国)有限公司，浪潮商用机器有限公司，OpenPower Foundation, OpenCAPI Consortium, 复旦大学微电子学院

**法律声明**

参加“2018 OpenCAPI+OpenPower 异构计算设计大赛”的所有参赛选手，请在提交参赛作品前仔细阅读本法律声明，如提交参赛作品，则视为完全理解、同意并接受本法律声明的全部条款。

参赛选手保证其提交的参赛作品为独立开发和原始创作，不侵犯任何第三方的知识产权及其它法律权益。参赛作品系指由参赛选手参与本次大赛产生的或在本次大赛下提交的评审论文。

所有参赛选手同意授予大赛主办方对参赛作品及其所有知识产权的许可，该许可包括发表、编辑、复制、发行、展示、翻译、汇编、通过信息网络传播参赛作品、基于参赛作品准备演绎作品以及分发(包括在内部的和向外部的）参赛作品及其演绎作品的副本、以及使用、生产、销售和以其他方式实施参赛作品。该许可包括允许大赛主办方通过第三方从事上述许可范围内的事项。上述许可为不可撤销的、非专有的、全球性的且无需支付许可费的许可。根据本条授权大赛主办方使用许可应包括其授予第三方可撤销或不可撤销的分许可的权利。

无论参赛选手有无过错，如参赛选手因参与本次大赛而导致主办方受到任何第三方权利指控，或使主办方遭受任何名誉、声誉或经济上的、直接或间接的损失，主办方有权要求参赛选手采取合理的措施，使主办方免受任何影响或消除、减小影响。并有权立即取消参赛选手参赛资格，或取消其已获得的奖项并要求返还奖品。主办方公司同时就其因此而遭受的任何名誉、声誉或经济上的、直接或间接的损失保留向参赛选手进一步索赔的权利。

“2018 OpenCAPI+OpenPower 异构计算设计大赛”的解释权归主办方所有。本法律声明根据中华人民共和国相关法律解释。

CAPI/OpenCAPI在线教程

<http://list.youku.com/albumlist/show/id_51706703.html?spm=a2h1n.8251843.0.0&&ascending=1>

2018 CAPI+OpenPower异构计算设计大赛

基于OpenPower的海洋物联网安全认证协议

设计与实现

学校： 哈尔滨工业大学

队名： 720第一分队

成员： 李峰

杨肇原

张懿

徐天亮

夏锦城

日期： 2018年8月14日

摘 要

海洋信息系统为各项海洋工作提供完整的信息支撑与安全畅通的信息通信渠道，最大限度地发挥海洋信息资源的价值。基于上述背景，本方案通过OpenPower平台实现一次性双向口令认证协议，解决海上信息采集临时组网通信安全问题；同时能够完成数据信息加密和解密功能，保障数据传输的安全问题。

本文详细描述了基于OpenPower平台实现双向安全认证协议的系统框架，涉及各种加密算法模块的硬件实现和软件调用调试，并对其安全性进行分析。并将安全认证协议在单片机上进行模拟实验，验证所设计的安全协议方案具有可行的。同时本文在对LBUS总线进行详细研究的基础上，深入学习能够处理大数据的异构算法OpenPower平台，研究该平台所支持的AXI总线协议，完成RSA、AES等加密算法模块封装成IP核，进而挂载到AXI总线的过程。实现安全认证协议在OpenPower平台中的调用，通过对总线上寄存器进行控制，实现基于AXI总线的 RSA加密模块的信息加密功能。

**关键字：**海洋组网；OpenPower平台；双向认证；加密算法

目 录

1 参赛选题 6

1.1 背景综述 6

1.2 方案论证与选择 7

2 方案描述 7

2.1 系统框架 8

2.2 安全认证协议设计 10

2.2.1 安全协议总体概述 10

2.2.2 注册阶段设计 11

2.2.3 认证与密钥协商阶段设计 12

2.3 加密模块设计 13

2.3.1 AES 加密模块设计 13

2.3.2 MD5 加密模块设计 15

2.3.3 RSA 加密模块设计 16

2.4 安全协议的可行性分析 19

3 测试计划 21

3.1 AXI 总线工作流程概述 21

3.2 RSA 模块在 AXI 总线的挂载 21

3.2.1 AXI 总线寄存器的申请 22

3.2.2 RSA 模块的 IP 核封装 22

3.2.3 IP 核的挂载 23

3.3 测试数据集 24

3.4 当前设计的资源占用情况 24

4 测试结果和分析 26

4.1 测试环境 26

4.2 测试结果 26

5 结语和展望 27

# 参赛选题

## 背景综述

海洋信息系统发展是国家海洋战略的重要组成部分。采集海洋的信息数据包括温度、密度、潮汐、水深等敏感信息, 这些信息在海面作业、海上航行和海洋军事领域起着尤为重要的作用。比如在军事领域方面，风向和海浪会影响重武器的发射精度；水文参数和地形信息影响作战计划的制定， 如果敌方获取了对应的参数，则会针对地形信息制定反策略计划， 严重危害我国海上领土安全。可见，保障海洋信息的安全尤为重要。

我国在党的十八大报告中对海洋安全提出的一系列要求，包括海洋资源开发、海洋环境保护、海洋经济发展和海洋权益维护等，为了实现我国建设海洋强国的目标，国家制定了一整套海洋战略规划和实施计划，其中涉及政治、军事、文化、经济、外交、管理、法律、科技、安全、社会、教育等相关战略领域。这些海洋战略的实施依赖于完整且安全的海洋信息系统。海洋信息向陆上数据中心的传输及各终端的协同观测，需组网完成。但与陆上网络不同，海洋信息的观测设备位于海上及水下，传输环境复杂。特别是水下，传输环境更为复杂多变，需要有针对性的研究相关技术。面对实际的应用需求，采集对应的海洋数据信息，此时需要在海上建立临时的无线网络进行互相传输和信息通信，整个结构实现监测船与空中、海上、海下等移动设备之间的相互通信。值得的注意的是海洋信息传输的基础是海洋通信技术，没有通信基础设施的支持，海洋信息不可能传输到陆上。但海洋通信环境存在特殊性，如海面通信节点的稀疏性，水天通信距离远(卫星)，水下传输环境的复杂多变性，给水下、水面、水天通信链路的可靠性带来了很大的挑战，需要考虑解决。

在物联网环境中，不断增长的终端设备给无线网络带来压力和考验。终端设备数量庞大而且无人看守，终端设备自身的安全无法保障；带有感知功能的终端设备能够获取信息，但同时恶意攻击者也会采取相同的手段截获信息。此外，恶意攻击者通过技术手段获取用户的个人隐私信息，从而给用户带来伤害。安全协议主要针对于信息传输安全层，保证信息传输过程中数据的机密性，完整性，真实性和新鲜性。为此，建立一套强大的安全认证体系，对于物联网的建设和发展来说，显得尤为重要。这就要求我们设计一种双向认证协议，保障海上监测船终端与无人机、无人船、水下探测机器人(AUV)等移动设备的通信安全。

在海洋观测网中, 如何确保信息在传输过程中不被他人篡改, 如何鉴别信息的完整性。对于完整性鉴别技术, 我们可以通过对身份、口令、密钥以及信息数据等项实施鉴别， 在通过加密技术实现对数据的保护。我国的海洋观测数据对于国防、国家安全、国民经济具有重要的作用，因此海洋观测数据传输的安全性也是必须要考虑研究的内容，包括基于硬件认证协议的身份鉴别技术、基于硬件加速的大数据加解密技术。

## 方案论证与选择

我国物联网现有安全认证机制主要包括以下几个方面，包括组认证机制， 设备认证机制，基于认证的秘钥协商机制等。在数字网络中的物联网设备正进行百万计的互联网通信，安全必须建立基于整个系统来建立。

在目前已有的双向认证协议中，林扬武提出一种基于 Hash 函数和消息认证码(MAC)的双向口令认证协议，尽管作者分析能够抵抗所有已知攻击，但存在密钥窃取泄露的危险。通过喻丽春提出的基于 AES 和 RSA 算法的一次性口令认证思想。结合两种方案的优点，提出一种新的基于 AES、RSA 和消息认证码的一次性双向口令认证协议。该协议使用 AES 对称加密和 RSA 非对称加密算法保护数据的传输安全，使用 Hash 函数和消息认证码(MAC)完成双向口令认证过程。同时为保证大数据处理的速度、性能及可靠性，最终确定选择在OpenCAPI+OpenPower平台上实现本文提出的双向认证协议，即使用CAPI共享接口技术和异构计算处理平台，从而达到性能和成本的最优化。

# 方案描述

本文的研究重点是面向海上临时组网的双向认证技术。使用 AES 对称加密和 RSA 非对称加密算法保护数据的传输安全，使用 Hash 函数和消息认证码(MAC)完成双向口令认证过程。在此基础上设计并实现相应的安全认证协议。

1. 设计安全认证协议，分步实施移动终端和服务器端的相互认证过程，并对设计好的安全认证协议进行安全性和可行性分析。

2. 研究OpenPower平台， 充分理解异构计算多核心处理器系统针对FPGA等特定任务的实现过程，同时研究CAPI技术，它是允许外部设备（I/O Device）和处理器CPU 共享内存的接口技术，由此解决大数据应用的加速问题。

3. 实现安全认证协议，将各模块封装成 IP 核，通过 AXI 总线对 IP 核进行数据的读写来完成安全认证协议。

本文的结构如图 2‑1所示。



图 2‑1 本文结构图

安全协议的实现主要基于OpenCAPI+OpenPower平台，该平台使用特有的共享接口技术完成处理器与FPGA之间的交互功能，实现整个系统的双向认证协议设计调试过程，处理大数据也可达到高速、高效的目的。

## 系统框架

本文通过OpenPower平台实现双向认证协议的系统框架图，如图 2‑2所示。

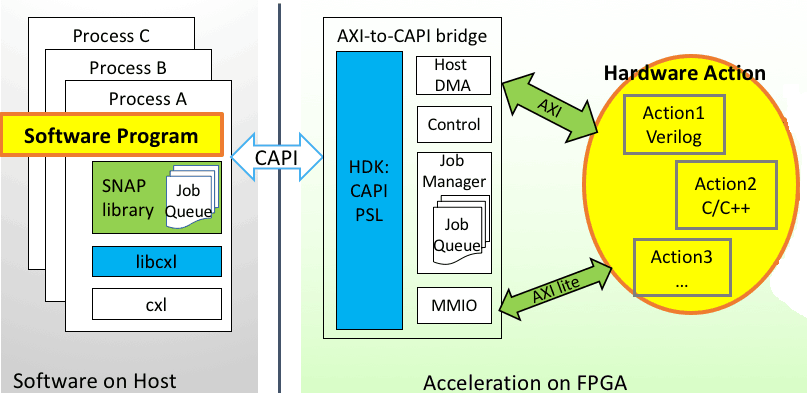


图 2‑2 系统架构图

1. 系统总体架构：主机（Host）<==CAPI==>FPGA；

2. 主机（Host）端: 运行软件平台为Ubuntu Linux；硬件环境：IBM-Power8系列CPU，通过PCIE总线与CAPI交互（包括控制，状态，数据）；

3. FPGA端：逻辑上包括两大部分：

a、实现分担CPU负荷的密集运算Action（图中黄色区域）；

b、IBM提供的接口：实现PCIE<==>AXI（以及AXI-lite）之间Data，control， status交互的 Framework（主要由PSL---power service layer实现）和SNAP CORE，其中SNAP CORE 包含5个套件：分别是MMIO， MMIO-2-AXImaster，Job Manager，Control 和 Host DMA。

其中，MMIO走AXI-lite总线，负责FPGA action的控制和状态设置；MMIO-2-AXImaster负责桥接AXImaster 与MMIO以及CAPI；Host DMA走AXI总线，即FPGA Action与Host Application交换数据的主要通道；Job Manager用于对诸于c\_job（SNAP CORE中负责连接）、 m\_job（FPGA的action）等实现同步执行等或其它多job 协调；control 负责一些内部控制。

4. 主机（Host）中运行Software Program（以下依IBM规范统称为Application），Application 通过对SNAP libaray 及其底层函数调用，实现PCIE与CAPI间交互，例如：实现host memory 数据透过CAPI与寄居于FPGA中的PSL架构的Framework，以及SNAP CORE 交互；

5. FPGA 外挂D-RAM Card （以上框图未有画出），当Action短时间内要求处理数据巨大时，可以考虑通过AXI总线接入FPGA；

本文基于OpenCAPI+OpenPower平台实现的双向安全通信认证协议的软硬件系统结构框图如图 2‑3所示。图中服务器与FPGA硬件逻辑之间通过CAPI接口实现数据的存储及传输，并使用AXI总线通过软件程序进行RSA、AES等加密模块的调试及调用，完成双向认证协议的基本功能。其中软件部分工作流程如图 2‑4所示，主要通过一直循环来实现硬件的相应调度。



图 2‑3 软硬件实现系统结构框图



图 2‑4 软件部分工作流程图

## 安全认证协议设计

本节主要阐述安全认证协议实施的具体步骤。对于每一次所使用的加密算法会做具体的阐述。由于验证信息加密及运算是整个认证协议的核心，而信息的传输则是根据实际情况具体设定，并且传输方式种类繁多，因此本节只着重进行认证协议的设计并且拟使用串口进行数据的传输。在认证协议完成之后，会对其进行安全性以及可行性分析，以验证双向安全认证的目的。

### 安全协议总体概述

完整的协议包括注册阶段和认证与密钥协商阶段两部分。约定以 ME 表示移动设备，这里的移动设备包括无人机、无人船、水下探测机器人(AUV) 等无线设备，MS 表示监测服务器，PWME 表示对应 IDME 的口令信息，RSA 算法生成的公钥和私钥分别以(e，N)和(d，N)表示，KP 表示 AES 对称算法生成的密钥，H(x)表示 x 的哈希值，Ee(x)表示(e，N)加密 x，EKP(x)表示使用 KP 加密x，MAC 值是通过 KP 加密的一组数据信息，“：”表示连接符号。完整安全认证协议的流程图如图 2‑5所示。



图 2‑5安全认证协议总体流程图

### 注册阶段设计

当移动设备 ME 要与监测服务器 MS 进行通信时，首先需要通过注册阶段，具体请求和密钥产生过程，如图 2‑6所示。



图 2‑6 注册阶段流程图

具体工作流程如下：

第 1 步：ME 向 MS 发送连接请求。

第 2 步：MS 通过 RSA 加密算法中密钥生成阶段，生成公钥(e，N)和私钥

(d，N)，发送公钥(e，N)到 ME 端。

第 3 步：ME 利用 MD5 算法计算口令信息 PWME 的 Hash 值 H(PWME)，在使用 MS 的公钥(e，N)加密 H(PWME)，发送 Ee(H(PWME))到 MS 端。

第 4 步： MS 使用私钥(d ， N) 解密得到 H(PWME)， 判断移动设备是否重复，如果不重复，保存该数据 H(PWME)，同时 MS 建立口令验证表，表中含有每个用户的 ID 信息以及通过该 ID 计算的 Hash 散列值。如果重复，则提示移动设备已经被使用，同时告知 ME 修改注册 ID。服务端返回注册成功。

### ****认证与密钥协商阶段设计****

认证与密钥协商阶段的前两步为密钥协商 ，后四步为双向认证过程， 完整流程图，如图 2‑7所示。



图 2‑7认证与秘钥协商阶段流程图

具体工作流程如下：

第 1 步：移动设备 ME 向监测服务器 MS 提交认证请求，同时 ME 使用AES 加密算法生成加密密钥 KP，并且生成一个随机数 RME，使用公钥(e，N)加密 KP，发送 Ee(KP)和 Ee(RME⊕H(PWME))到 MS 端。

第 2 步：MS 使用私钥(d，N)进行解密，得到密钥 KP，计算得到 RME，生成随机数 RMS，使用 KP 加密 RMS， RME，发送 EKP(RMS)和 EKP(H(RME))到 ME 移动设备端。

第 3 步：ME 使用 KP 解密得到 RMS，并验证 EKP(H(RME))。此时 ME 调用密钥 KP 计算 MAC 的值 KP(H(PWME)，RMS， RME，IDME)，然后将 IDME 与计算出来的 MAC 值使用 MS 的公钥(e，N)加密后，发送到 MS 端。

第 4 步 ：MS 使 用 私 钥 (d，N) 解 密 得 到 KP(H(PWME)，RMS，RME，IDME)，根据 IDME 查口令验证表得到 H(PWME)， 调用密钥 KP 计算 H(PWME)， RMS，RME ， IDME 的 MAC 值， 并检验该值与移动设备 ME 传过来 MAC 值是否一致。如果一致，则监测服务器 MS 认证移动设备 ME 成功，协议继续；若不一致，则验证失败，拒绝移动设备 ME 的认证请求。

第 5 步： MS 调用密钥 KP 计算 RMS， RME， IDME 的 MAC 值， 并将该值KP(RMS，RME，IDME)使用 MS 的私钥(d，N)加密后，发送给移动设备 ME 端。

第 6 步： ME 使用公钥(e，N) 解密得到 KP(RMS，RME，IDME)，并调用密钥 KP 计算 RMS，RME，IDME 的 MAC 值，并检验该值与收到的 MAC 值是否一致。如果一致，则移动设备 ME 认证监测服务器 MS 成功，协议完成；否则拒绝监测服务器 MS 的认证请求。

## 加密模块设计

### AES 加密模块设计

AES 是一个新的可以用于保护电子数据的加密算法。明确地说，AES 是一个迭代的、对称密钥分组的密码，它可以使用 128、192 和 256 位密钥，并且用 128 位（16 字节）分组加密和解密数据。与公共密钥密码使用密钥对不同，对称密钥密码使用相同的密钥加密和解密数据。通过分组密码返回的加密数据的位数与输入数据相同。迭代加密使用一个循环结构，在该循环中重复置换（permutations）和替换(substitutions）输入数据。置换是数据的重新排列， 而代替是用一个单元数据替换另一个。AES 使用了几种不同的技术来实现置换和替换。AES 加密流程图如图 2‑8 所示。



图 2‑8 AES 加密流程图

基于 AES 加密流程图，拟设计 AES 加密模块如图 2‑9所示。



图 2‑9 AES 加密模块

AES 加密模块中，各端口对应的信息及名称如表 2‑1所示。

表 2‑1 AES 模块端口对应信息

|  |  |  |  |
| --- | --- | --- | --- |
| 传输信息方向 | 名称 | 含义 | 位数 |
| In | Clk | 时钟 | 1 |
| In | Enable | 使能 | 1 |
| In | Reset | 复位 | 1 |
| In | Key\_ready | 密钥准备传输 | 1 |
| In | Data\_ready | 明文准备传输 | 1 |
| In | Key | 密钥 | 256 |
| In | Data | 明文 | 256 |
| Out | State | 加密状态 | 1 |
| Out | Key\_end | 密钥传输结束 | 1 |
| Out | Data\_out\_ready | 密文准备传输 | 1 |
| Out | Data\_out | 输出密文 | 256 |
| Out | Finish | 结束信号 | 1 |

### MD5 加密模块设计

MD5 算法是典型的消息摘要算法，属 Hash 算法一类。MD5 算法对输入任意长度的消息进行运行， 产生一个 128 位的消息摘要。其前身有 MD2 、MD3 和 MD4 算法，它由 MD4、MD3 和 MD2 算法改进而来。不论是哪一种MD 加密算法，它们都需要获得一个随机长度的信息并产生一个 128 位的信息摘要。如果将这个 128 位的二进制摘要信息换算成十六进制，可以得到一个32 位的字符串，因此加密完成后的 16 进制的字符串长度为 32 位。MD5 加密算法整体流程图如图 2‑10所示。



图 2‑10 MD5 算法整体流程图

基于 MD5 加密算法流程，拟设计 MD5 加密模块如图 2‑11所示。



图 2‑11 MD5 加密模块

MD5 加密模块中，各端口对应的信息如表 2‑2所示。

表 2‑2 MD5 模块端口对应信息

|  |  |  |  |
| --- | --- | --- | --- |
| 传输信息方向 | 名称 | 含义 | 位数 |
| In | Clk | 时钟 | 1 |
| In | Enable | 使能 | 1 |
| In | Reset | 复位 | 1 |
| In | Data\_ready | 明文准备传输 | 1 |
| In | Data | 明文 | - |
| Out | Data\_out\_ready | 密文准备传输 | 1 |
| Out | Data\_out | 密文 | 128 |
| Out | State | 加密状态 | 1 |

### RSA 加密模块设计

RSA 加密方式在安全认证协议中被多次使用，故对RSA加密算法模块进行详细的阐述与设计。RSA加密算法是一种[非对称加密算法](https://baike.baidu.com/item/%E9%9D%9E%E5%AF%B9%E7%A7%B0%E5%8A%A0%E5%AF%86%E7%AE%97%E6%B3%95)。在[公开密钥加](https://baike.baidu.com/item/%E5%85%AC%E5%BC%80%E5%AF%86%E9%92%A5%E5%8A%A0%E5%AF%86)密和[电子商业](https://baike.baidu.com/item/%E7%94%B5%E5%AD%90%E5%95%86%E4%B8%9A)中RSA 被广泛使用。对极大整数做[因数分解](https://baike.baidu.com/item/%E5%9B%A0%E6%95%B0%E5%88%86%E8%A7%A3)的难度决定了RSA算法的可靠性。换言之， 对一极大整数做因数分解愈困难，RSA算法愈可靠。假如有人找到一种快速因数分解的算法的话，那么用RSA加密的信息的可靠性就肯定会极度下降。但找到这样的算法的可能性是非常小的。今天只有短的RSA钥匙才可能被强力方式解破。到目前为止，世界上还没有任何可靠的攻击RSA算法的方式。只要其钥匙的长度足够长，RSA加密的信息实际上是不能被解破的。RSA算法加密与解密过程如下所示：

RSA加密过程：

（1）A 提取消息m的消息摘要h(m)，并使用自己的私钥对摘要h(m)进行加密，生成签名s。

（2）A将签名s和消息m一起，使用B的公钥进行加密，生成密文c，发送给用户B。

RSA 解密过程：

（1）B接收到密文c，使用自己的私钥解密c得到明文m和数字签名s。

（2）使用A的公钥解密数字签名s解密得到H(m)。

（3）使用相同的方法提取消息m的消息摘要h(m)。

（4）比较两个消息摘要，相同则验证成功，不同则验证失败。

RSA 算法流程图如图 2‑12所示。



图 2‑12 RSA 算法流程图

在 RSA 加密算法中，私钥和公钥的生成方式如图 2‑13所示。

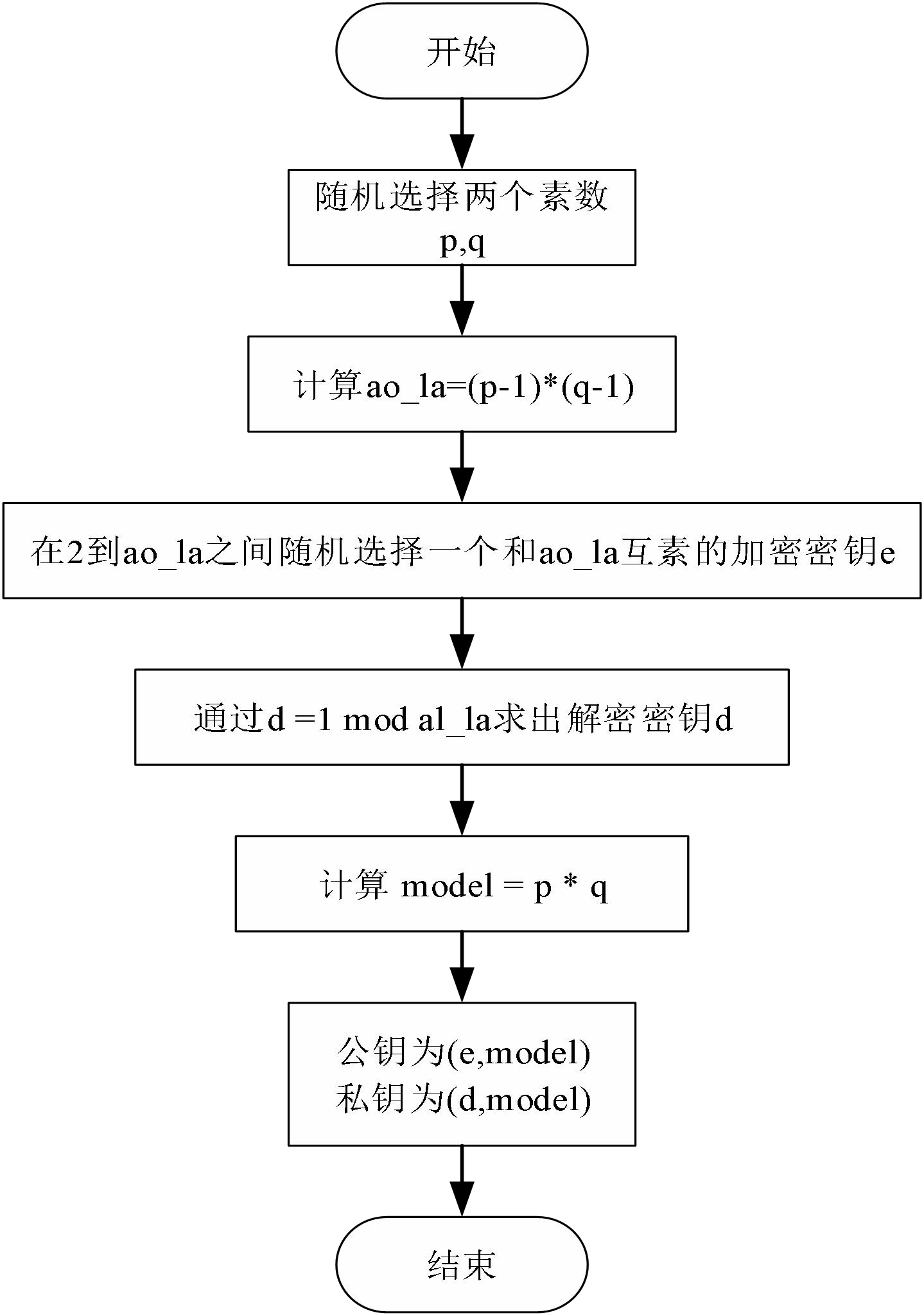


图 2‑13 RSA 公钥与私钥生成方式

根据RSA加密流程图，对RSA加密模块进行设计，RSA加密模块如图 2‑14所示。



图 2‑14 RSA加密模块

RSA加密模块中，各端口对应的信息如表2-3所示。

表 2‑3 RSA模块端口对应信息

|  |  |  |  |
| --- | --- | --- | --- |
| 传输信息方向 | 名称 | 含义 | 位数 |
| In | Key | 密钥值 | 1024 |
| In | Mod | 模值 | 1024 |
| In | Data | 明文值 | 1024 |
| In | Key\_ready | 密钥准备传输 | 1 |
| In | Mod\_ready | 模值准备传输 | 1 |
| In | Data\_ready | 数据准备传输 | 1 |
| Out | Finish | 传输完成信号 | 1 |
| Out | Data\_out\_ready | 数据准备传输 | 1 |
| Out | Data\_out | 密文值 | 1024 |
| Out | Mod\_end | 模值传输完成 | 1 |
| Out | Key\_end | 密钥传输完成 | 1 |
| Out | State | 状态 | 2 |

在RSA模块中，所有的高电平代表动作的开始。State表示RSA模块中状态机的工作状态，有初始化、工作两个状态。

## 安全协议的可行性分析

安全协议的可行性分析，通过完成通信安全协议在单片机上的实现，验证本文提出的一种基于AES和RSA算法与消息认证码(MAC)组合的一次性双向口令认证协议方案可行。

通过运用两个STM32单片机，分别做为移动端和服务器终端，进行安全协议的搭建，在安全协议认证完成之后，可以进行数据的传输。单片机使用串口进行通信，在PC端使用超级终端对整个加密过程进行监控，以观测每一步加密之后所得的值。如图 2‑15所示。

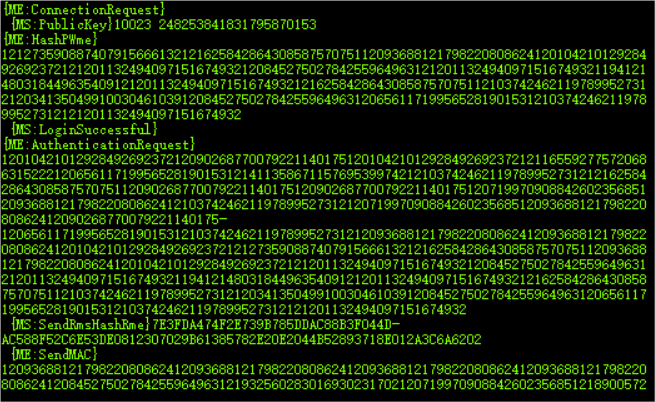


图 2‑15加密过程监控

作为移动端的单片机设有3个按键，分别为通信开始，数据1，数据2。按动通信开始按键时，两个单片机顺次进行整个通信交互过程，当安全认证完成后，可以在移动端通过按键来传输数据，传输的结果会在做为服务器端的单片机上进行显示。

每个单片机都使用中断的工作方式，当串口有数据时就进行数据的处理。由于产生的随机数不同，故每次认证成功之后值均不同。认证成功如图 2‑16所示。

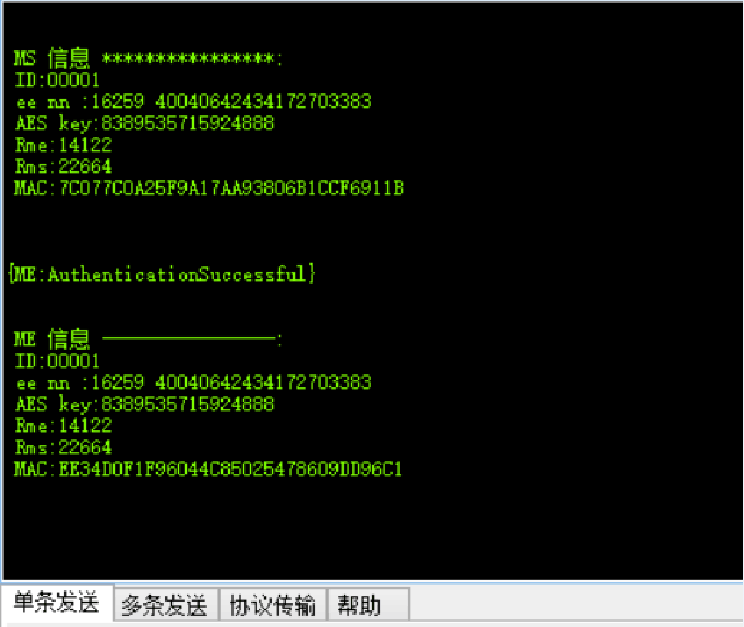


图 2‑16 认证成功

# 测试计划

上一章具体阐述了安全认证协议的设计及其安全性分析，本章重点阐述在OpenCAPI+OpenPower平台上，基于AXI总线来完成安全认证协议的实现。由于整个安全认证协议的搭建就是将各个模块进行IP核的封装，然后将各个 IP 核挂载到 AXI 总线上， 然后通过CPU进行控制。故本章重点研究将 RSA 算法挂载到 AXI 总线上，并将挂好的RSA算法进行验证，证明其加密功能的完整性。本章后续描述以Xilinx Zynq为例，介绍IP核与系统的互联，随着项目进行，会将Zynq的部分替换为OpenCAPI/CAPI的开发环境SNAP。

## AXI 总线工作流程概述

将模块挂载在AXI总线上，实际上就是在 AXI 总线上定义寄存器，即将AXI总线上已有的寄存器，按照实际需要引出。Vivado会自动根据用户定义好的寄存器将IP核封装好。然后将生成的电路进行综合，实现，然后烧写进ZedBoard开发板。为了实现对封装的 IP 核进行控制，用户在SDK中编写C语言程序，通过对寄存器所在的地址，进行数据的输入与输出。用户生成的IP核与SDK程序之间的关系如图 3‑1所示。

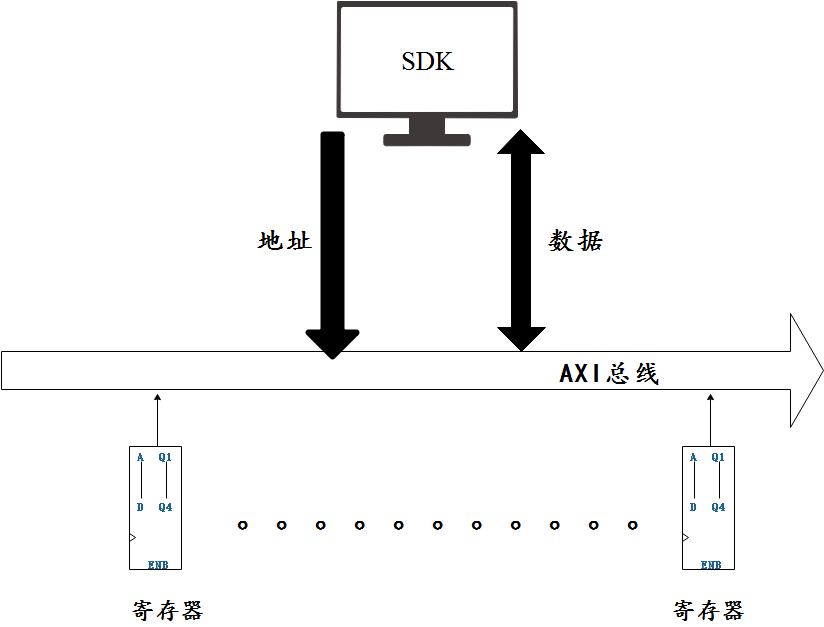


图 3‑1 IP 核与 SDK 程序关系

## RSA 模块在 AXI 总线的挂载

首先对 RSA 模块进行测试，得出 RSA 模块的时序图如图 3‑2 所示。

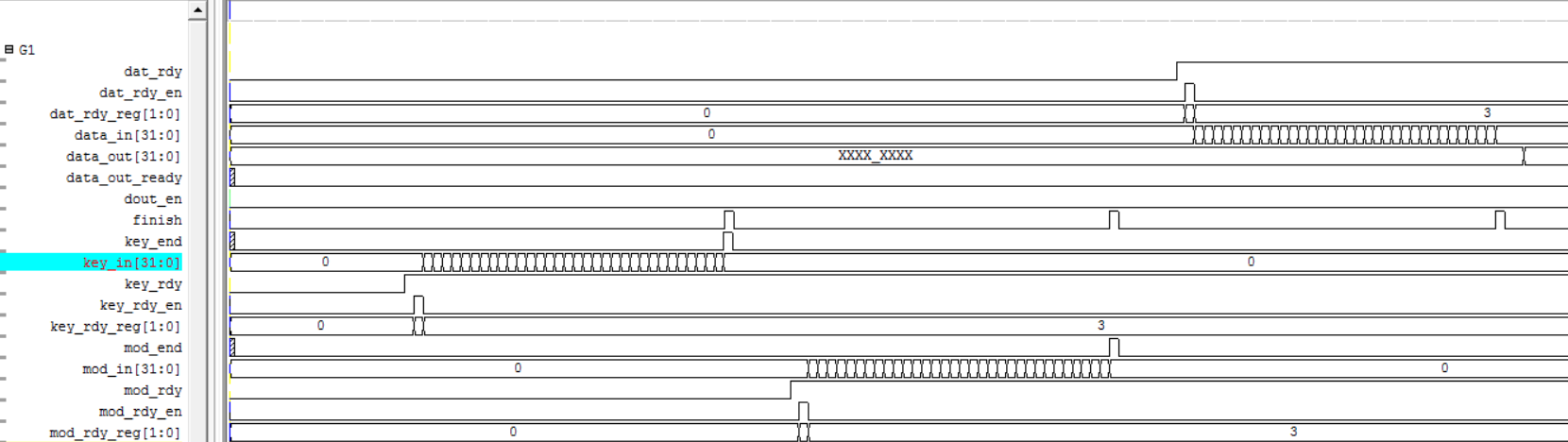


图 3‑2 RSA模块时序图

由时序图可知，数据进入 RSA 模块的次序是密钥，模值，数据值。每当1024 位的数据进入模块之前，其对应的准备完成信号会先产生脉冲，然后对应的使能端信号拉高，同时在寄存器中写入 1。1024 位的数据会在 32 个周期内以每周期 32 位的速度完成传输。在每 1024 位数据传输完成后，finish 信号会对应产生脉冲。

### AXI 总线寄存器的申请

基于 RSA 模块的时序， 拟向总线申请 35 个 32 位寄存器 slv\_reg0 至slv\_reg34 ， slv\_reg 寄 存 器 为 总 线 上 可 读 可 写 寄 存 器 。 其 中 slv\_reg0 至slv\_reg31 为数据寄存器，key\_ready 信号存储于 slv\_reg32[0], mod\_ready 信号存储于 slv\_reg32[1], data\_ready 信号存储于 slv\_reg32[2]。加密完成信号存储于 slv\_reg33[0]。RSA 模块复位信号存储slv\_reg34[0]。这样，可以通过查找对应的寄存器中的数据来查看工作状态或者直接对对应的寄存器进行数据的读入读出工作。

由于 AXI 总线上的寄存器最多只支持 32 位的数据，且寄存器数量最多为512 个，故不能将 1024 位的数据直接传输。为了使所用的寄存器数量尽可能的小，故数据寄存器重复使用，当收到 finish 信号后，直接将 1024 数据送入RSA 模块，32 个寄存器等待下一个 1024 位数据的输入。

### RSA 模块的 IP 核封装

在向总线申请寄存器完成之后，只需在 AXI 总线协议中添加对应的用户协议即可。不同于基于 AXI 总线的呼吸灯等事例，RSA 模块既要从总线中读取数据，又要将数据送回到总线上，而呼吸灯只需要从总线读取数据即可。故呼吸灯中，可以将用户逻辑写于顶部模块，在底部模块中只引出寄存器即可。而由于 RSA 模块对数据有交互的功能，故将 RSA 模块的用户逻辑写于底部模块中，外部则可以对寄存器进行直接的数据读写。

用户逻辑中详细向总线声明数据的送入方式，由于有验证的需要，故对RSA 模块进行一定的处理。在 ZedBoard 开发板上申请 5 个 LED 灯，用于实时监测 RSA 模块的工作状态。其中，led1，led2，led3 分别用于验证密钥，模值，和明文的传输状态。Led4，led5 用于监测 RSA 模块的工作状态。

将在第2节设计完成的 RSA 模块直接添加至所封装 IP 核的顶层逻辑之中， 只需在用户逻辑中对 RSA 模块进行例化，即可完成对该 RSA 模块的调用。封装完成的 RSA\_IP 核如图 3‑3所示。封装完成的 IP 核可以直接挂载在AXI 总线上。

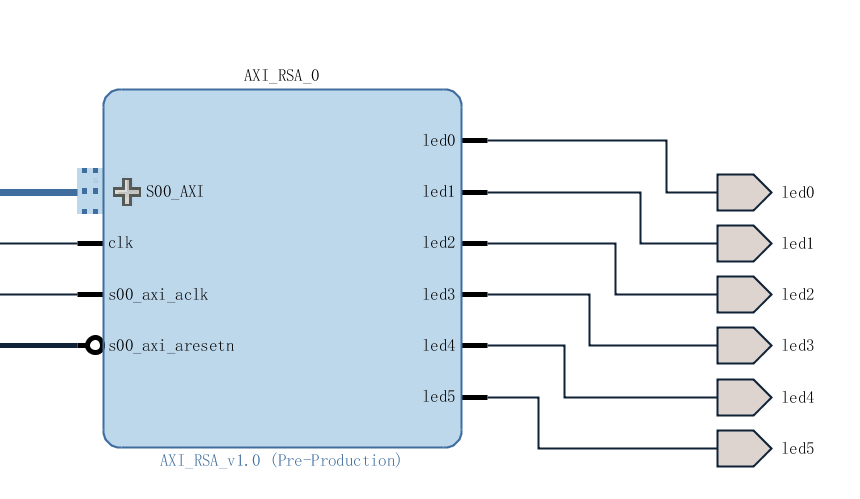


图 3‑3 封装完成的 RSA\_IP 核

### IP 核的挂载

当 IP 核封装完成之后，在 Vivado 的 Block Design 界面开始搭建完整实现 RSA 加密的系统。首先添加控制核（以ZYNQ核为例），本次搭建的系统未调用 GPIO 模块。在对 Vivado 下达自动连接的指令之后，Vivado 会自动生成整个系统，并自动添加复位模块和互连模块。完整的系统图如图 3‑4所示。

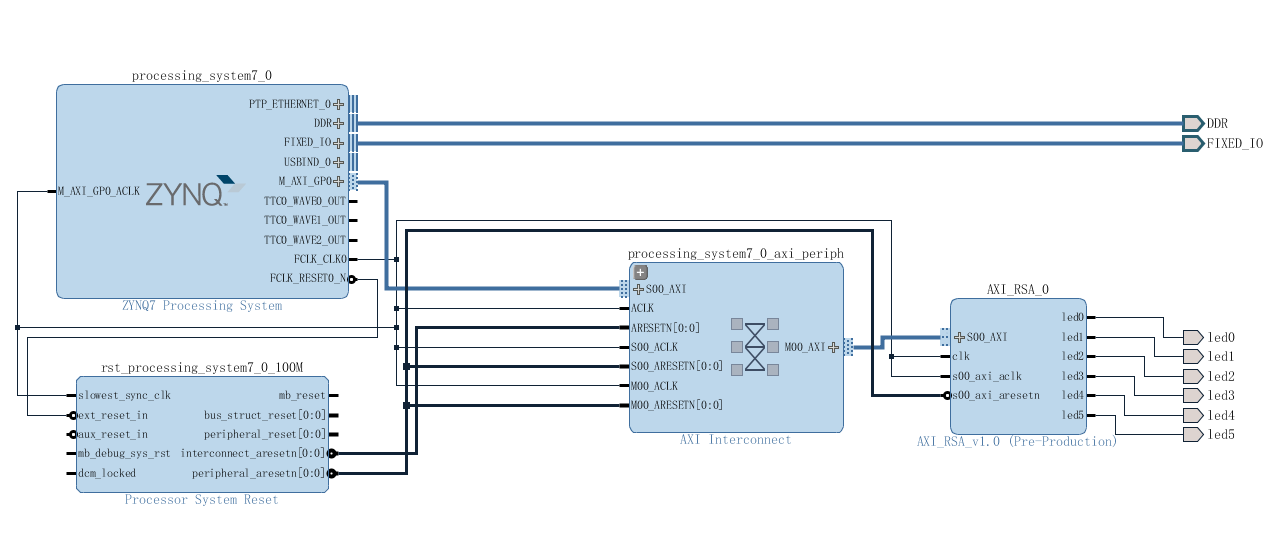


图 3‑4 完整的 RSA 系统流程图

在系统搭建完成后，进行综合、实现、生成bit流。将bit流烧进ZedBoard，即可进行后续的在SDK中的开发工作。当SDK模块程序完成之后，即可实现 PL 与 PS 的交互，即软件与硬件的结合。进而完整的实现 RSA 算法。

其他的模块也类似于 RSA 算法的 IP 核封装，在 AXI 总线上依次挂载， 在 SDK 中对寄存器的调用进行设计，即可实现完整的安全认证协议。

## 测试数据集

表3‑1 测试数据集

|  |  |
| --- | --- |
| Item | Description |
| AES | 实现AES算法的测试验证 |
| MD5 | 实现MD5算法的测试验证 |
| RSA | 实现RSA算法的测试验证 |
| IP核的封装 | 实现RSA等算法的IP核封装验证 |
| IP核的挂载 | 实现RSA等算法的IP核在AXI总线的挂载验证 |
| SDK程序设计 | 实现对PS部分寄存器的调用验证 |
| FPGA测试 | 实现整体功能在FPGA上的测试验证 |
| OpenCAPI调试 | 实现整体功能在OpenPower平台上的测试验证 |

主要完成RSA、AES等加密算法模块的AXI总线挂载，通过OpenCAPI接口，在OpenPower 平台上实现各个模块之间的调度调试，验证双向认证协议的功能。

## 当前设计的资源占用情况

针对当前设计，完成RSA加密算法模块的AXI总线挂载后，用Vivado综合工具可以对设计用到的LUT、DSP、IO等资源的占用情况进行查看。在Synthesis>Report Utilization界面可以查阅，如图3-5所示。

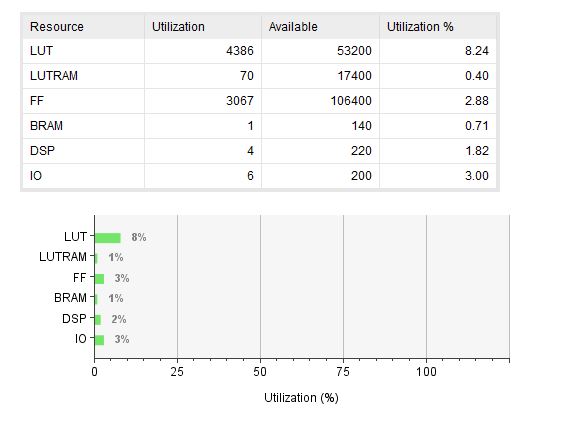


图 3‑5资源占用率

本设计对资源的占用较少。在Vivado综合工具中以ZYNQ为例查看资源占用情况，可以发现，LUT可用量为53200，占用量为4386，占比为8.24%；LUTRAM可用量为17400，仅占用了0.40%。FF的占用率为2.88%，BRAM的占用率为0.71%，DSP的占用率为1.82%，IO的占用率为3.00%。除了LUT占用率接近10%，其余占用率均在5%以下，LUTRAM和BRAM占用率更是在1%以下，可以看出本设计对资源的要求是很低的，即便不使用高端的开发平台，一些中低端的开发平台已然足够使用，可以大大降低成本，使得本设计具有低成本、高性能的突出优点。据此在后期设计将AES、MD5等算法挂载到AXI总线上，可以通过合理的优化，使得整体设计满足低成本、高性能的要求。

# 测试结果和分析

## 测试环境

## 测试结果

# 结语和展望

本文针对面向海洋物联网的安全认证协议进行研究，包括：安全认证协议的设计，安全认证协议的安全性及可行性分析，安全认证协议在OpenCAPI+OpenPower平台上的实现。主要设计思想如下：

（1）设计了安全认证协议，在协议中分别运用了RSA，AES，MD5算法来对传输的信息进行加密。详细阐述了在认证的每一步中所传输的信息，尽可能的使认证协议的安全性增加。

（2）实现了安全认证协议的可行性分析，通过在单片机上实现双向安全通信协议，验证还方案具有可行性，并研究了OpenCAPI+OpenPower平台以及该平台所支持的AXI总线，并对AXI总线的工作方式进行了学习。

基于此，本文提出将RSA模块挂载到AXI总线上，并对其功能进行了测试，验证本文所设计的安全认证协议的一部分。

对于本研究课题的工作展望如下：

本文针对认证协议的安全性进行充分的设计，但是并未考虑实际的移动终端的内存大小以及计算的速度。未来工作可以在考虑实际情况的条件下安全认证协议进行一定的优化，达到内存和计算速度的最优解。

本文在实现安全认证协议时，只优先考虑将RSA模块挂载到AXI总线上，并对其功能进行了验证。未来工作可以通过RSA模块的挂载方法，将认证协议中的AES等其他算法都封装成IP核并挂载到AXI总线上，在SDK中编写程序，达到数据的计算以及存储的功能，进而在OpenCAPI+OpenPower平台上实现完整的安全认证协议。