**xilinx加密方案实现及验证说明V1.0**

|  |  |
| --- | --- |
| 拟 制 |  |
| 审 核 |  |
| 会签 |  |
| 批 准 |  |

**修订记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **修订版本** | **日期** | **作者** | **修改描述** | **备注** |
| V1.0 | 2023.08.11 | 黄成伟 | 初始版本 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1 引言 2](#_Toc141202412)

[1.1 目的 2](#_Toc141202413)

[1.2 范围 2](#_Toc141202414)

[1.3 缩略语定义 2](#_Toc141202415)

[1.4 参考资料 2](#_Toc141202416)

[2 需求概述 2](#_Toc141202417)

[2.1 功能需求 3](#_Toc141202418)

[2.2 接口需求 3](#_Toc141202419)

[2.3 性能需求 3](#_Toc141202420)

[2.4 边界需求 3](#_Toc141202421)

[3 移植说明 3](#_Toc141202422)

[3.1 移植工作简介 3](#_Toc141202423)

[3.1.1 函数的远程调用及核间参数的传递原理说明 3](#_Toc141202424)

[3.1.2 移植所需完成的工作 4](#_Toc141202425)

[3.2 移植的具体实现 5](#_Toc141202426)

[3.2.1 A53\_0核代码调整内容 5](#_Toc141202427)

[3.2.2 A53\_3核代码的调整 7](#_Toc141202428)

[4 附件 9](#_Toc141202429)

# 引言

## 目的

本文用于说明xilinx加密方案实现及验证过程，供项目组开发人员和软件维护人员阅读。

## 范围

本文档只限于塞防科技项目组研发、测试、产品以及项目相关人员作为内部信息对齐使用，未经公司批准以及书面授权不允许任何人以任何形式对本文档复制、传播、改动。

## 缩略语定义

|  |  |  |
| --- | --- | --- |
| **缩略语** | **全称** | **描述** |
|  |  |  |
|  |  |  |
|  |  |  |

## 参考资料

|  |  |
| --- | --- |
| **名称** | **版本** |
|  |  |
|  |  |
|  |  |
|  |  |

# 需求概述

1. 采用xilinx加密方案，实现对应用程序的加密和加密后程序的正常运行

## 功能需求

本方案不涉及到新功能的开发

## 接口需求

本方案不涉及到新的接口设计

## 性能需求

五

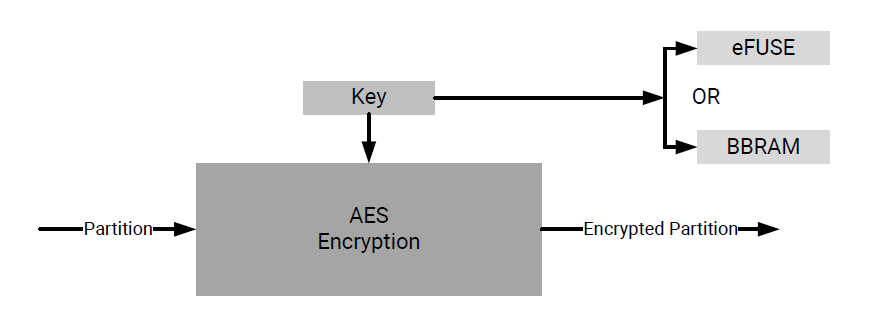
## 边界需求

无

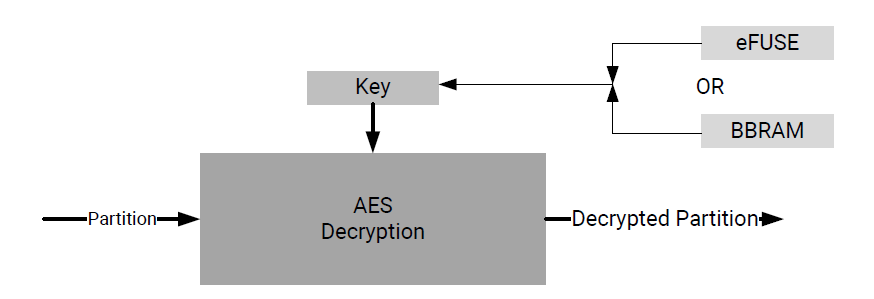
# xilinx加密方案

## xilinx加密方案原理

### 加密过程



### 解密过程



## 加密相关文件的准备

使用xilinx的加密方案需要准备下列内容：AES key，AES key烧写程序，使用AES key加密的Boot升级程序，使用AES key加密的应用程序

### AES key的生成

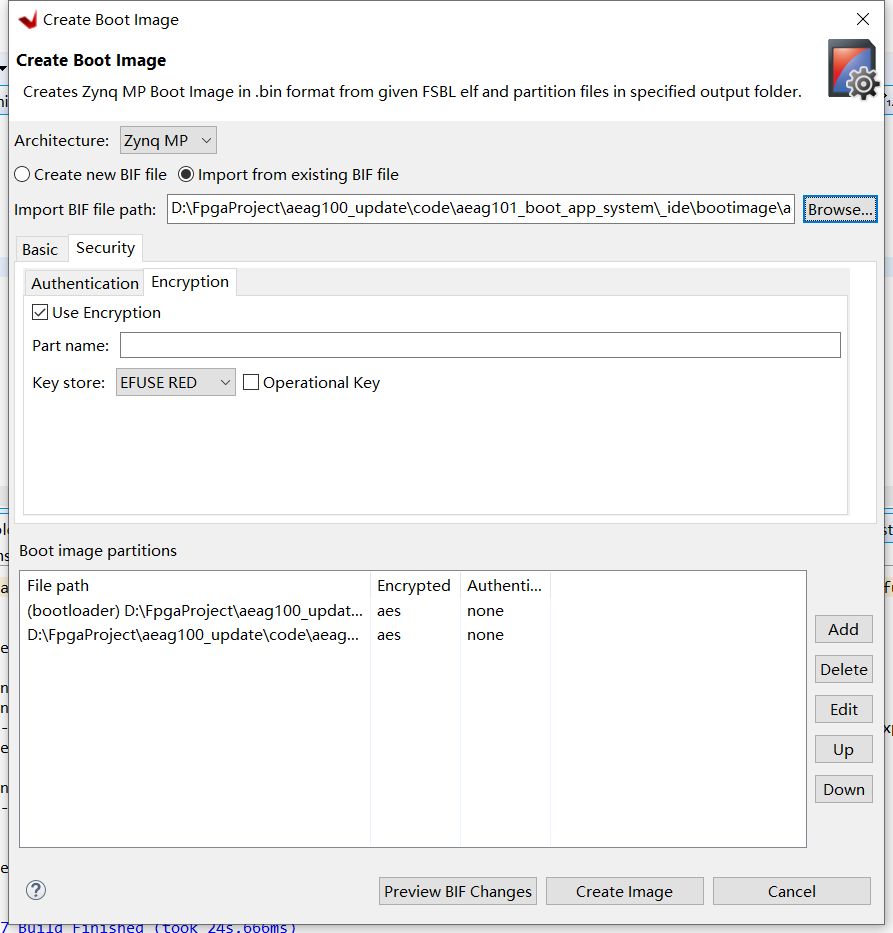
可以选用不同的工具生成AES key，长度为256bit。

### AES key烧写程序

程序项目的生成和编译参见文档《xapp1319-zynq-usp-prog-nvm.pdf》的Programming the AES Key in BBRAM、Programming eFUSEs for AES and RSA Cryptographic Functions两节，编译项目前需要用上一步生成的AES key代替原头文件中的AES key，在本方案中使用eFUSEs保存AES key，eFUSEs只能烧写一次。

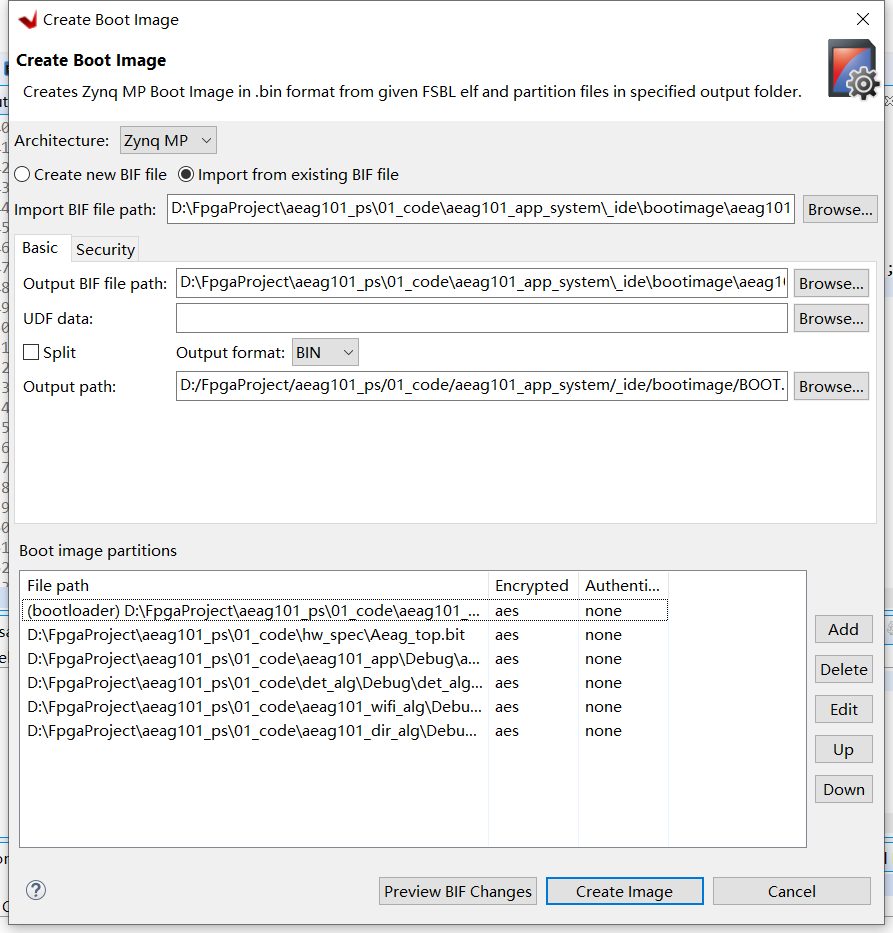
### 使用AES key加密Boot升级程序

使用<https://adasgitlab.autel.com/aeag/aeag100_upgrade.git>中的项目，生成boot程序后，使用vitis的create boot image生成boot的image文件，image的所有文件均进行加密，文件加密时需要选取key文件（key文件格式参加压缩包xapp1319-zynq-usp-prog-nvm.zip），不同文件key文件的内容相同、key同AES key烧写程序中的key一致，但key文件名不同，fsbl（bootloader）文件必须加密，界面如下所示：



### 使用AES key加密APP程序

APP项目编译完成后，同样使用vitis的create boot image生成APP的image文件，image的所有文件均进行加密，文件加密时需要选取key文件，不同文件key文件的内容相同、key同AES key烧写程序中的key一致，但key文件名不同，fsbl（bootloader）文件必须加密，需要注意，如果生成的APP image文件大于16M，有可能不能启动，界面如下所示：



## 加密方案验证

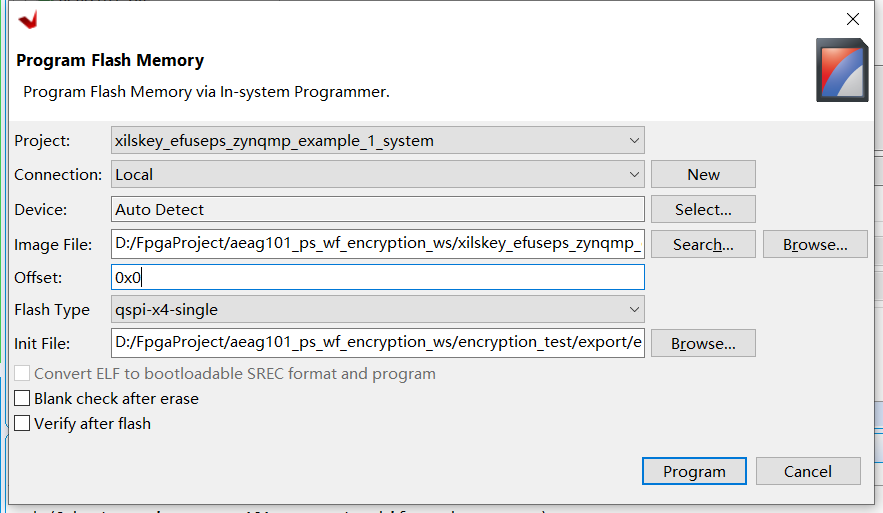
按照下列步骤验证加密程序是否正常运行：

### 把key写入设备的eFUSEs

eFUSE只能烧写一次，烧写成功后再次烧写会提示烧写失败。把key写入设备的eFUSE可以使用两种方式：

第一种方式：把编译好的AES key烧写程序通过JTAG口下载到设备内存运行，串口会打印运行结果。

第二种方式：把生成的AES key烧写程序在vitis内通过JTAG口烧写到设备FLASH，烧写成功后，设备重新上电，串口会打印运行结果。

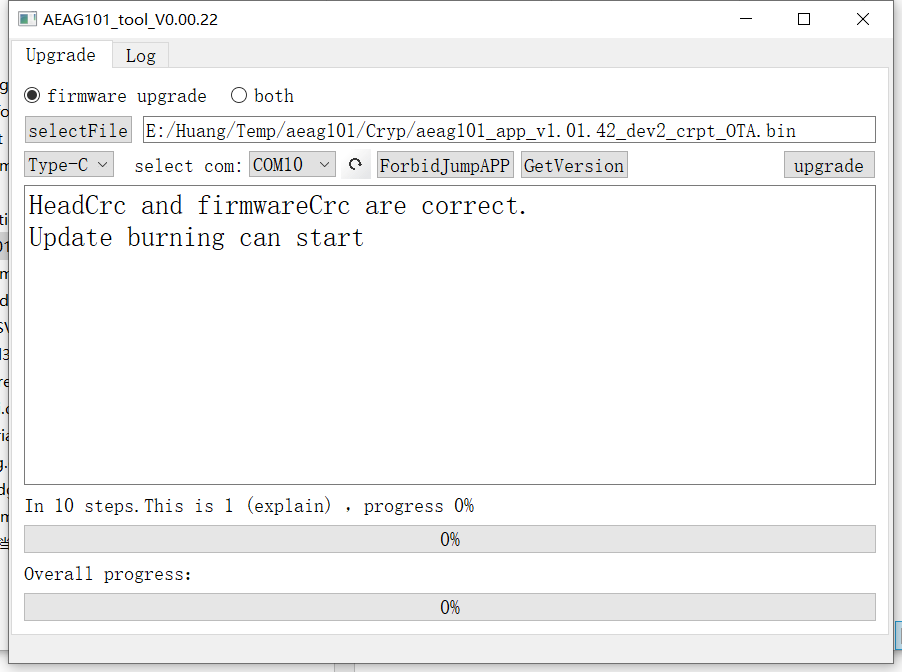


### AES key加密Boot升级程序

把AES key加密的boot程序在vitis内通过JTAG口烧写到设备FLASH，烧写成功后，设备重新上电，串口会打印运行结果，并提示可以烧写APP程序。

### 烧写AES key加密后的APP程序

使用公司自研的上位机烧写程序烧写AES key加密后的APP程序到flash，烧写成功后，设备重新上电，设备如果正常工作，且版本号应该同烧写程序的版本号一致，说明烧写的加密软件成功运行。



# 附件

无。