



# Hi3519AV100/Hi3556AV100 安全启动使用指南

文档版本 00B02  
发布日期 2018-09-04

版权所有 © 深圳市海思半导体有限公司 2018。保留一切权利。

未经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



**HISILICON**、海思和其他海思商标均为深圳市海思半导体有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受海思公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，海思公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

## 深圳市海思半导体有限公司

地址：深圳市龙岗区坂田华为基地华为电气生产中心 邮编：518129

网址：<http://www.hisilicon.com>

客户服务电话：+86-755-28788858

客户服务传真：+86-755-28357515

客户服务邮箱：[support@hisilicon.com](mailto:support@hisilicon.com)



# 前言

## 概述

本文档主要介绍 Hi3519AV100/Hi3556AV100 安全启动的使用方法，主要内容包括：安全启动介绍、安全镜像生成步骤及 OTP 烧写说明。

支持如下启动介质：SPI NOR FLASH、SPI NAND FLASH、并口 NAND FLASH 和 eMMC。



说明

未有特殊说明，Hi3556AV100 与 Hi3519AV100 内容一致。

## 产品版本

与本文档相对应的产品版本如下。

| 产品名称    | 产品版本 |
|---------|------|
| Hi3519A | V100 |
| Hi3556A | V100 |

## 读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

## 修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。



| 修订日期       | 版本    | 修订说明                       |
|------------|-------|----------------------------|
| 2018-09-04 | 00B02 | 第 2 次临时版本发布。<br>1.4 小节涉及修改 |
| 2018-08-08 | 00B01 | 第 1 次临时版本发布。               |



# 目 录

前 言.....i

1 安全启动介绍.....1

    1.1 普通安全 boot 镜像结构.....1

    1.2 加密安全 boot 镜像结构.....3

    1.3 安全启动流程.....4

    1.4 安全启动源代码目录说明.....4

2 安全镜像生成.....7

    2.1 安全 U-boot 生成步骤.....7

    2.2 密钥文件介绍.....8

3 OTP 烧写步骤.....9



# 插图目录

图 1-1 普通安全 boot 镜像结构图..... 2

图 1-2 加密安全 boot 镜像结构图..... 3

图 1-3 安全启动流程 ..... 4



# 1 安全启动介绍

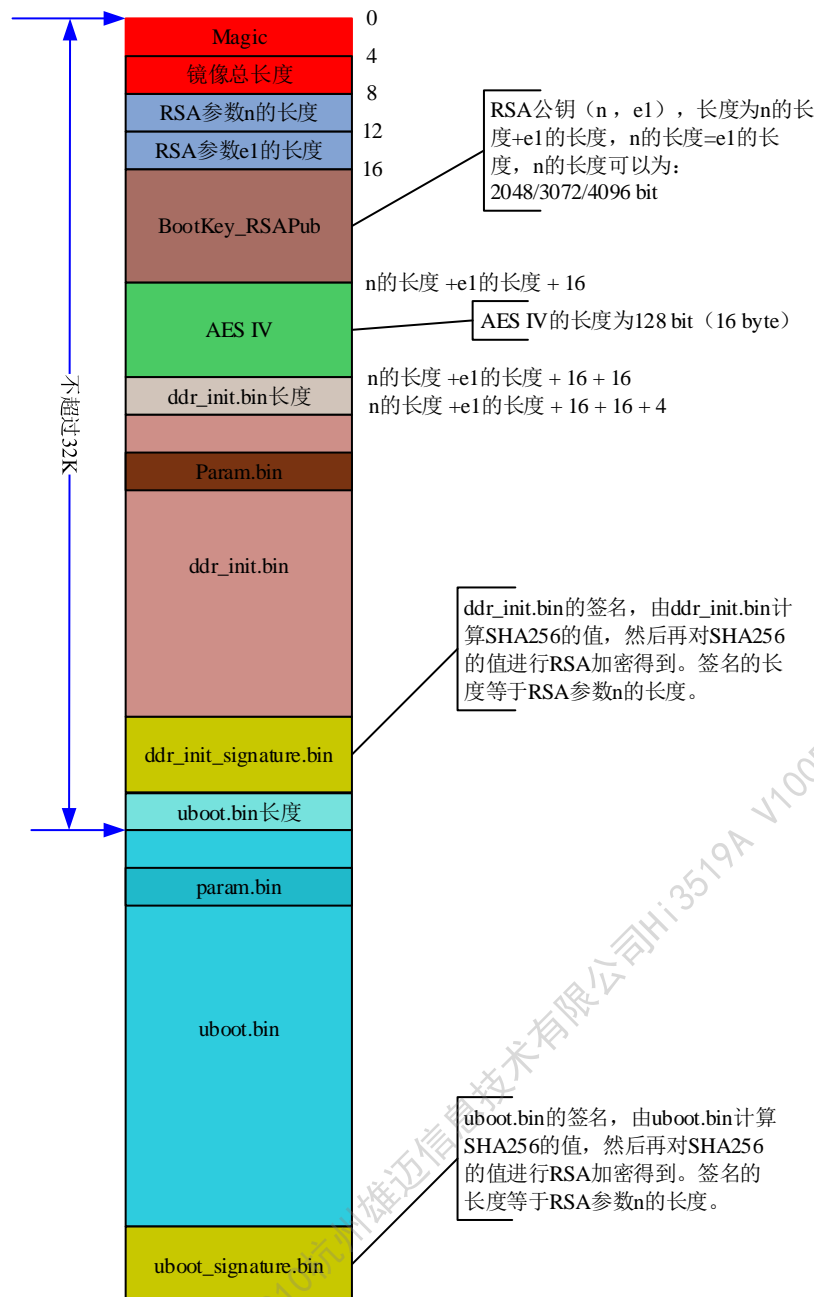
Hi3519AV100 支持普通安全 boot 启动和加密安全 boot 启动，其差异在于普通安全 boot 镜像中，`ddr_init.bin` 和 `u-boot.bin` 是明文；加密的安全 boot 镜像中，`ddr_init.bin` 和 `u-boot.bin` 是密文。

## 1.1 普通安全 boot 镜像结构

Hi3519AV100 普通安全 U-boot 镜像结构如[图 1-1](#) 所示。



图1-1 普通安全 boot 镜像结构图



普通安全启动 uboot 镜像由公钥镜像、ddr\_init.bin 镜像（包括 param.bin 和 ddr\_init.bin）、非安全 uboot.bin 镜像和 ddr\_init.bin 的数字签名（ddr\_init\_signature.bin）、非安全 uboot.bin 的数字签名（uboot\_signature.bin）及它们各自的长度信息组成。

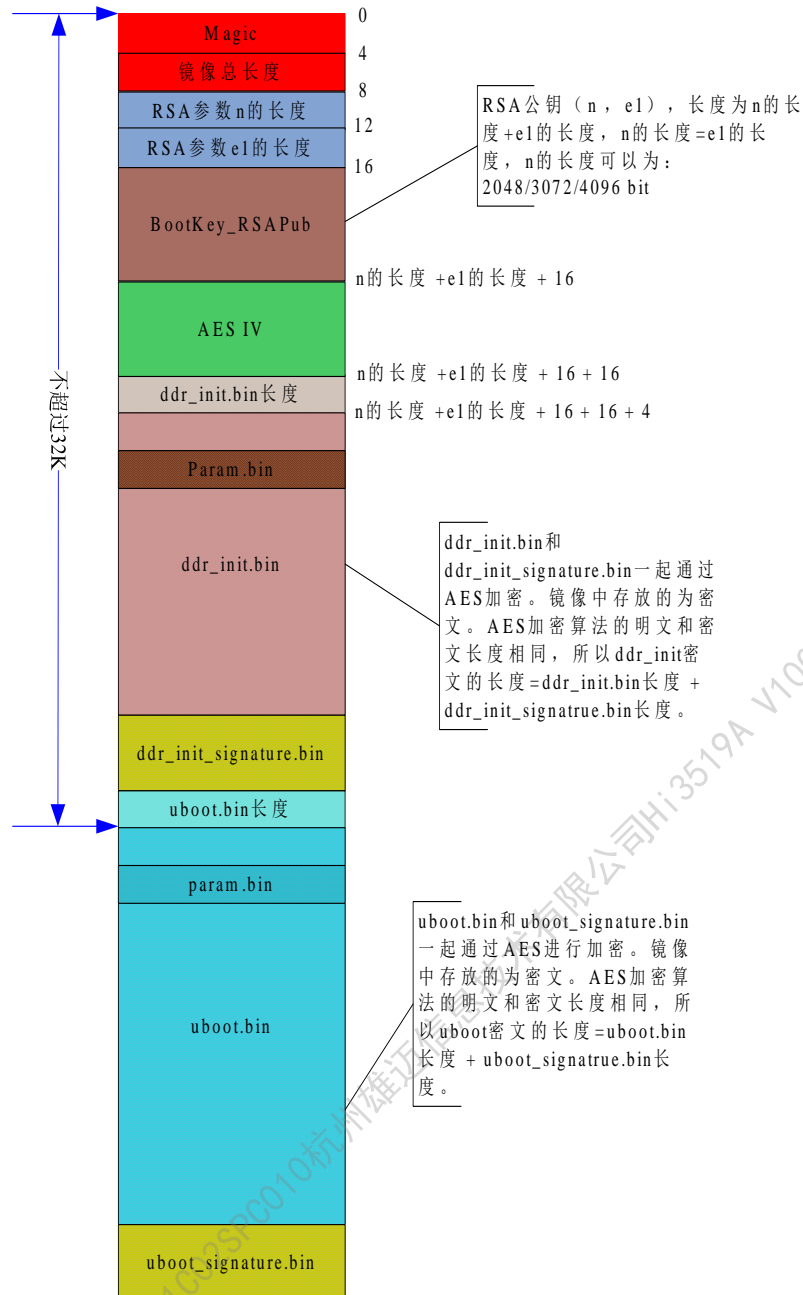
其中：RSA 支持 2048、3072 和 4096 两种格式。AES IV 的值为 0。





## 1.2 加密安全 boot 镜像结构

图1-2 加密安全 boot 镜像结构图



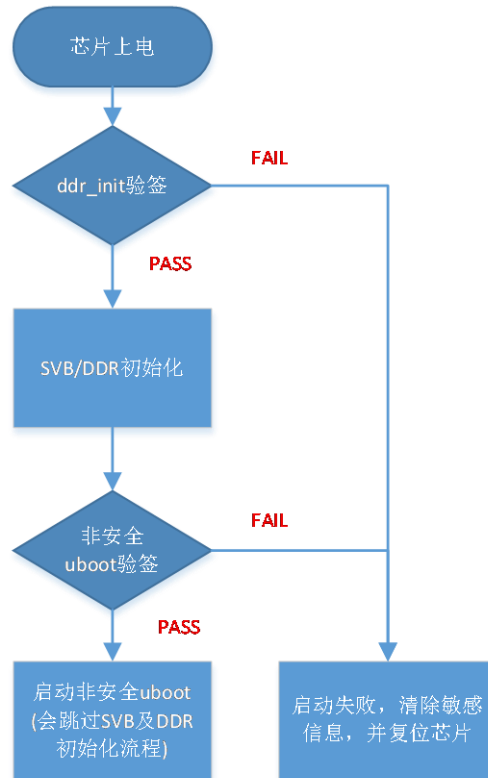
加密安全启动 uboot 镜像由公钥镜像、ddr\_init.bin 镜像（包括 param.bin 和 ddr\_init.bin）及其数字签名通过 AES 加密后的密文、非安全 uboot.bin 镜像及其数字签名通过 AES 加密后的密文和它们各自的长度信息组成。

其中：RSA 支持 2048、3072 和 4096 两种格式。AES IV 的值为非 0 值。



## 1.3 安全启动流程

图1-3 安全启动流程



## 1.4 安全启动源代码目录说明

安全启动源代码目录为 hi3519av100\_secureboot\_release，其详细目录结构如下：

```
|—— CASignTool -> CASignTool_Linux_BVT/CASignTool/bin/CASignTool_m64
|—— CASignTool_Linux_BVT ----- CASignTool 工具源代码目录
|   |—— build.sh
|   |—— CASignTool
|   |—— libCASign
|   |—— readme.txt
|—— ddr_init ----- DDR 初始化源代码目录
|   |—— boot
|   |—— cfg.mk
|   |—— ddr_init_hi3519av100.bin
```



|  |  |                          |  |
|--|--|--------------------------|--|
|  |  | drv                      |  |
|  |  | include                  |  |
|  |  | linker.lds               |  |
|  |  | linker.lds.mk            |  |
|  |  | Makefile                 |  |
|  |  | mkddrinit.sh             |  |
|  |  | reg_info.bin             | ----- DDR 初始化表格，编译时，需要换成最新版本。  |
|  |  | ddr_init_hi3519av100.bin | ----- 生成的 DDR 初始化镜像。   |
|  |  | HASH                     | ----- 由 hash_modify.c 文件生成的 RSA Pub Key hash 值解析工具。  |
|  |  | hash_modify.c            |  |
|  |  | Makefile                 | ----- 安全启动发布包总 Makefile。   |
|  |  | rsa2048pem               | ----- 长度为 2048 Bit 的密钥文件存放目录。  |
|  |  | rsa2048pem.sh            | ----- 生成长度为 2048 Bit 密钥的脚本。  |
|  |  | rsa3072pem               | ----- 长度为 3072 Bit 的密钥文件存放目录。  |
|  |  | rsa3072pem.sh            | ----- 生成长度为 3072 Bit 密钥的脚本。  |
|  |  | rsa4096pem               | ----- 长度为 4096 Bit 的密钥文件存放目录。  |
|  |  | rsa4096pem.sh            | ----- 生成长度为 4096 Bit 密钥的脚本。  |
|  |  | secure_boot.cfg          | ----- 该文件用于设置 AES Key 和 IV 值，ddr_init.bin 和 u-boot.bin 镜像执行 AES 加密；若 AES Key 和 IV 值为空，则不执行 ddr_init.bin 和 u-boot.bin 镜像加密。 |
|  |  | sha256.cfg               | ----- 该文件设置执行的算法，已设置为 SHA256，不需要修改。  |
|  |  | u-boot-hi3519av100.bin   | ----- 非安全 uboot 镜像。  |



### 注意

安全启动流程中，执行的是 hi3519av100\_secureboot\_release 目录下的 SVB 及 DDR 初始化代码，UBOOT 中的 SVB 及 DDR 初始化流程不会被执行！

因此在安全启动场景下，如果需要更新 SVB 或 DDR 初始化流程，须修改如下目录文件：

osdrv/opensoruce/uboot/hi3519av100\_secureboot\_release/ddr\_init/drv/  
| cmd\_bin



- |—— cmd\_ddr\_training\_v2.c
- |—— ddr\_cmd\_ctl.c
- |—— ddr\_cmd\_loc.S
- |—— ddr\_ddrc\_v500.h
- |—— ddr\_ddrc\_v510.h
- |—— ddr\_ddrc\_v520.h
- |—— ddr\_ddrt\_s40.h
- |—— ddr\_ddrt\_t12\_v100.h
- |—— ddr\_ddrt\_t16.h
- |—— ddr\_ddrt\_t28.h
- |—— ddr\_interface.h
- |—— ddr\_phy\_s40.h
- |—— ddr\_phy\_t12\_v100.h
- |—— ddr\_phy\_t12\_v101.h
- |—— ddr\_phy\_t16.h
- |—— ddr\_phy\_t28.h
- |—— ddr\_training\_boot.c
- |—— ddr\_training\_console.c
- |—— ddr\_training\_ctl.c
- |—— ddr\_training\_custom.c
- |—— ddr\_training\_custom.h
- |—— ddr\_training\_impl.c
- |—— ddr\_training\_impl.h
- |—— ddr\_training\_internal\_config.h
- |—— Makefile

osdrv/opensource/uboot/hi3519av100\_secureboot\_release/ddr\_init/lowlevel\_init\_v300.c

发布包下，安全 BOOT 中 SVB、DDR 初始化流程同非安全 UBOOT 中 SVB、DDR 初始化流程保持一致。



## 2 安全镜像生成

### 2.1 安全 U-boot 生成步骤

步骤 1. 生成非安全 U-boot 镜像：

参考《Hi3519AV100/Hi3556AV100 U-boot 移植应用开发指南》中 “移植 U-boot” 章节。

步骤 2. 解压安全 U-boot 发布包：

```
tar xvf hi3519av100_secureboot_release.tgz
```

将步骤 1 生成的非安全 U-boot 镜像 u-boot-hi3519av100.bin 拷贝至 hi3519av100\_secureboot\_release 目录。

步骤 3. 拷贝 reg\_info.bin 文件：

从 osdrv/tools/pc/uboot\_tools 目录拷贝新生成的 reg\_info.bin 文件到 hi3519av100\_secureboot\_release/ddr\_init 目录。

步骤 4. 配置 secure\_boot.cfg 文件中的 KEY 和 IV：

如果要生成普通安全启动 uboot 镜像，secure\_boot.cfg 文件中的 KEY 和 IV 的值都设置为空；如果要生成加密安全启动 uboot 镜像，secure\_boot.cfg 文件中的 KEY 和 IV 的值需要设置。如：KEY=67452301efcdab8967452301efcdab89  
IV=01233210012332100123321001233210

步骤 5. 编译安全启动 uboot 镜像：

```
cd hi3519av100_secureboot_release
```

执行 make rsa2048pem\_gen、make rsa3072pem\_gen 或 make rsa4096pem\_gen

最终在 hi3519av100\_secureboot\_release 目录下生成对应的安全镜像。

----结束



### 注意

发布包脚本会在第一次编译时产生公钥和私钥文件，后续编译的安全镜像均采用第一次生成的公钥和私钥，如果要更新公钥和私钥，需手动删除 rsa2048pem、rsa3072pem 或 rsa4096pem 目录下的文件。

## 2.2 密钥文件介绍

```
├── rsa2048pem
│   ├── rsa2048_pem_hash_val.txt //文本格式的公钥 HASH 和寄存器配置命令
│   ├── rsa_priv_2048.pem        //PEM 格式私钥
│   ├── rsa_pub_2048.bin         //二进制格式公钥
│   ├── rsa_pub_2048.pem        //PEM 格式公钥
│   └── rsa_pub_2048_sha256.txt  //文本格式的公钥的 HASH 值
├── rsa3072pem
│   ├── rsa3072_pem_hash_val.txt //文本格式的公钥 HASH 和寄存器配置命令
│   ├── rsa_priv_3072.pem        //PEM 格式私钥
│   ├── rsa_pub_3072.bin         //二进制格式公钥
│   ├── rsa_pub_3072.pem        //PEM 格式公钥
│   └── rsa_pub_3072_sha256.txt  //文本格式的公钥的 HASH 值
├── rsa4096pem
│   ├── rsa4096_pem_hash_val.txt //文本格式的公钥 HASH 和寄存器配置命令
│   ├── rsa_priv_4096.pem        //PEM 格式私钥
│   ├── rsa_pub_4096.bin         //二进制格式公钥
│   ├── rsa_pub_4096.pem        //PEM 格式公钥
│   └── rsa_pub_4096_sha256.txt  //文本格式的公钥的 HASH 值
```



# 3 OTP 烧写步骤

步骤 1. 烧写非安全 U-boot，并启动 U-boot 至命令行；

步骤 2. 公钥 HASH 烧写（必选）：

```
mw 0x040A0008 0x6
mw 0x040A000c 0xxxxxxxxx
mw 0x040A0010 0xxxxxxxxx
mw 0x040A0014 0xxxxxxxxx
mw 0x040A0018 0xxxxxxxxx
mw 0x040A001c 0xxxxxxxxx
mw 0x040A0020 0xxxxxxxxx
mw 0x040A0024 0xxxxxxxxx
mw 0x040A0028 0xxxxxxxxx
```



说明

以上公钥 HASH 配置命令，可从 rsa2048\_pem\_hash\_val.txt 、rsa3072\_pem\_hash\_val.txt 或 rsa4096\_pem\_hash\_val.txt 中直接 copy。

```
mw 0x040A0000 0x2
mw 0x040A0004 0x1acce551
```

步骤 3. DDR 加扰 BIT 烧写（可选）：

```
mw 0x040A0034 0x1
mw 0x040A0030 0x2
mw 0x040A0000 0x4
mw 0x040A0004 0x1acce551
```

步骤 4. 安全启动 BIT 烧写（必选）：

```
mw 0x040A0034 0x0
mw 0x040A0030 0x1
```



```
mw 0x040A0000 0x4
mw 0x040A0004 0x1acce551
```

步骤 5. AES KEY 烧写（可选）：

```
mw 0x040a0008 0x0
mw 0x040a000c 0xxxxxxxxx
mw 0x040a0010 0xxxxxxxxx
mw 0x040a0014 0xxxxxxxxx
mw 0x040a0018 0xxxxxxxxx
```



说明

以上配置的 AES KEY 的值，需要与 secure\_boot.cfg 文件中的 KEY 值一致，写入时需要进行大小端转换。例如：secure\_boot.cfg 文件中“KEY=67452301efcdab8967452301efcdab89”，则上面设置寄存器的命令为“mw 0x040a000c 0x01234567;mw 0x040a0010 0x89abcdef; mw 0x040a0014 0x01234567; mw 0x040a0018 0x89abcdef;”

```
mw 0x040a0000 0x2
mw 0x040a0004 0x1acce551
```

步骤 6. 通过 U-boot 命令烧写安全镜像至启动介质，或通过 hitool 工具烧写安全镜像至启动介质。

----结束



注意

以上每个烧写步骤都必须谨慎小心，以免烧写错误导致芯片不可用。

如果要生成普通安全 uboot 镜像，secure\_boot.cfg 文件中的 KEY 和 IV 都要设置为空。