



Hi3861V100 / Hi3861LV100 Boot 移植应用

开发指南

文档版本 02

发布日期 2020-06-05

版权所有 © 上海海思技术有限公司2020。保留一切权利。

非经本公司书面许可，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海思技术有限公司的商标。

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。

上海海思技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编：518129

网址： <https://www.hisilicon.com/cn/>

客户服务邮箱： support@hisilicon.com



前言

概述

本文档描述了Hi3861V100/Hi3861LV100 RomBoot、LoaderBoot及FlashBoot工作流程，用户可参考此文档对FlashBoot进行二次开发。

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
Hi3861	V100
Hi3861L	V100



读者对象

本文档主要适用于以下工程师：



- 技术支持工程师
- 软件开发工程师

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
 危险	表示如不可避免则将会导致死亡或严重伤害的具有高等级风险的危害。
 警告	表示如不可避免则可能导致死亡或严重伤害的具有中等级风险的危害。



符号	说明
 注意	表示如不避免则可能导致轻微或中度伤害的具有低等级风险的危害。
须知	用于传递设备或环境安全警示信息。如不避免则可能会导致设备损坏、数据丢失、设备性能降低或其它不可预知的结果。 “须知”不涉及人身伤害。
 说明	对正文中重点信息的补充说明。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害信息。

修改记录

文档版本	发布日期	修改说明
02	2020-06-05	新增“ 5.3.2 用户根公钥配置 ”小节。
01	2020-04-30	第一次正式版本发布。 <ul style="list-style-type: none">更新“前言”的概述。在“1 Boot简介”中更新关于Boot分类的描述；新增图1-1。更新“2.1 下载镜像及烧写EFUSE”的描述。新增“3 LoaderBoot功能说明”小节。更新“4.2 Boot目录结构”标题名称；更新表4-1。更新“5.3.1 签名及加密配置”的图5-4。更新“5.3.3 签名工具介绍”的图5-5；新增-u、-f、-z参数的含义说明。更新“5.4 Boot可用API”标题名称。
00B 02	2020-02-12	更新“ 表4-1 ”的芯片固化接口头文件目录名、链接文件目录名、驱动源文件目录包含的内容。 新增“ 5.1 FlashBoot编译 ”、“ 5.3 FlashBoot安全启动配置说明 ”小节。
00B 01	2020-01-15	第一次临时版本发布。



目录

前言.....	i
1 Boot 简介.....	1
2 RomBoot 功能说明.....	3
2.1 下载镜像及烧写 EFUSE.....	3
2.2 检验及引导 Flashboot.....	3
3 LoaderBoot 功能说明.....	5
4 FlashBoot 说明.....	6
4.1 FlashBoot 启动流程.....	6
4.2 Boot 目录结构.....	6
5 FlashBoot 二次开发指南.....	9
5.1 FlashBoot 编译.....	9
5.2 调整内存布局.....	9
5.3 FlashBoot 安全启动配置说明.....	11
5.3.1 签名及加密配置.....	11
5.3.2 用户根公钥配置.....	12
5.3.3 签名工具介绍.....	12
5.3.4 密钥文件说明.....	13
5.4 Boot 可用 API.....	14



1 Boot 简介

Hi3861V100/Hi3861LV100 Boot分4部分：RomBoot、FlashBoot、LoaderBoot、CommonBoot。

RomBoot功能包括：

- 加载LoaderBoot到RAM，进一步利用LoaderBoot下载镜像到Flash、烧写EFUSE。
- 校验并引导FlashBoot。FlashBoot分为AB面，A面校验成功直接启动，校验失败会去校验B面，B面校验成功会修复A面再引导启动，否则复位重启。

FlashBoot功能包括：

- 升级固件。
- 校验并引导固件。

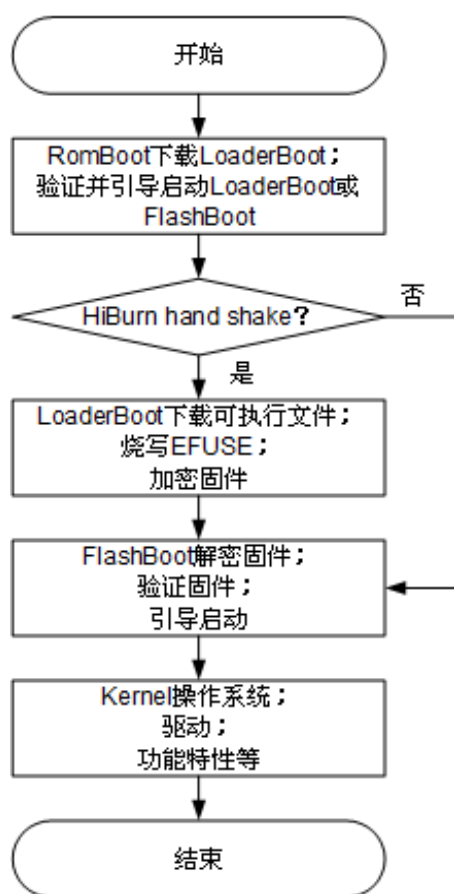
LoaderBoot功能包括：

- 下载镜像到Flash。
- 烧写EFUSE（例如：安全启动/Flash加密相关密钥等）。

CommonBoot为Flashboot与LoaderBoot共用的功能模块。



图 1-1 Boot 启动流程





2 RomBoot 功能说明

2.1 下载镜像及烧写EFUSE

2.2 检验及引导Flashboot

2.1 下载镜像及烧写 EFUSE

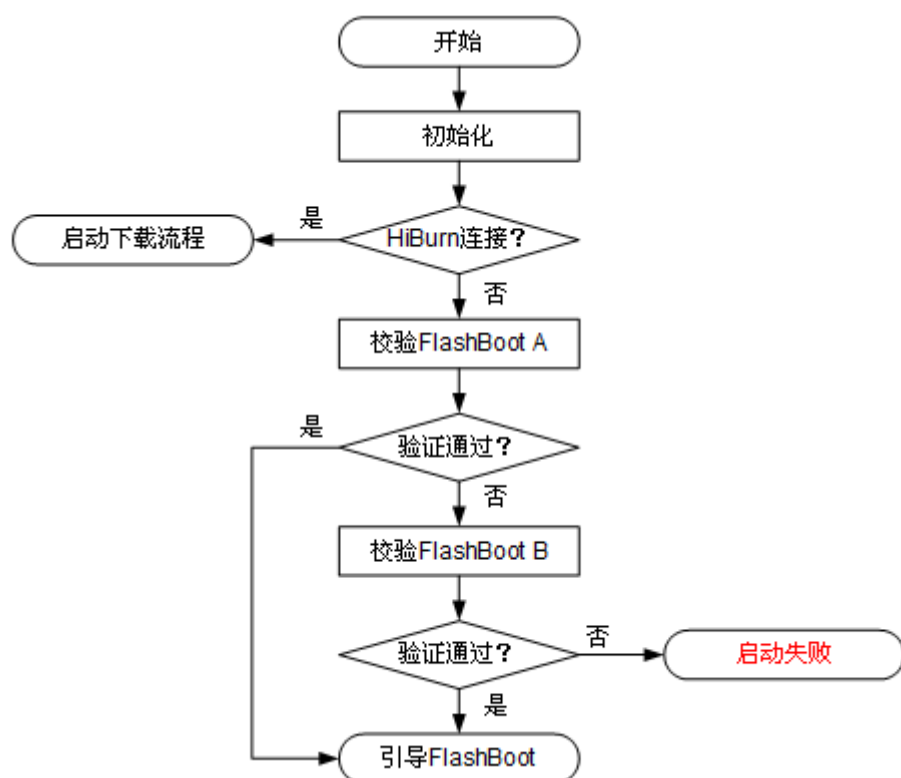
RomBoot通过加载Loaderboot实现下载镜像到Flash及烧写EFUSE的功能，具体操作请参见《Hi3861V100 / Hi3861LV100 HiBurn工具 使用指南》。

2.2 检验及引导 Flashboot

校验并引导FlashBoot流程如[图2-1](#)所示。



图 2-1 校验并引导 FlashBoot 流程图





3 LoaderBoot 功能说明

LoaderBoot是直接和HiBurn进行交互的组件，RomBoot无法直接实现烧写的功能，需要将LoaderBoot加载到RAM后，跳转到LoaderBoot，进一步通过LoaderBoot完成相关内容的烧写，LoaderBoot可烧写的内容包括：

- FlashBoot
- EFUSE参数配置文件
- 固件镜像（包括NV参数）
- 产测镜像

说明

LoaderBoot一般不涉及二次开发，如果有应用场景需要修改，可直接修改LoaderBoot源码，SDK会默认编译并更新LoaderBoot，LoaderBoot的目录结构请参见“[4.2 Boot目录结构](#)”。



4 FlashBoot 说明

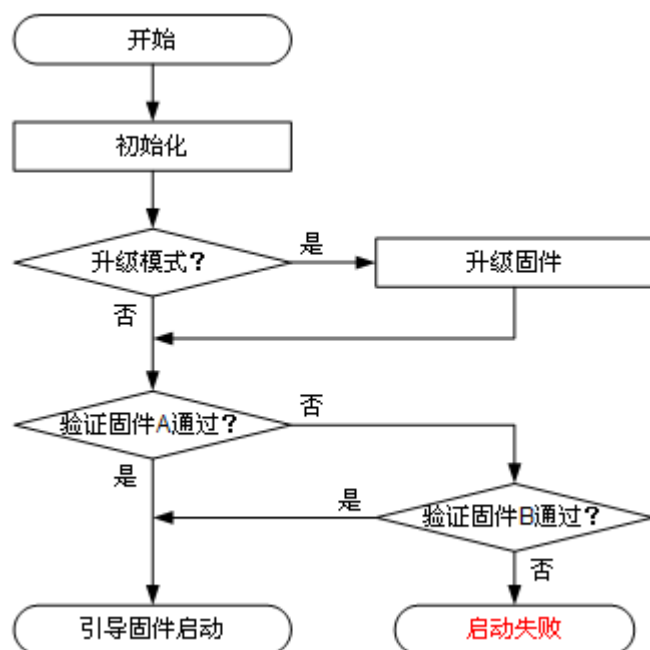
4.1 FlashBoot启动流程

4.2 Boot目录结构

4.1 FlashBoot 启动流程

校验并引导固件流程如[图4-1](#)所示。

图 4-1 校验并引导固件流程图



4.2 Boot 目录结构

SDK的Boot目录下为Boot的源码及头文件，主要目录结构如[表4-1](#)所示。



表 4-1 Boot 目录说明

目录名	路径	说明
flashboot	boot\flashboot\include	FlashBoot头文件目录。
	boot\flashboot\startup	启动汇编及主程序入口目录。
	boot\flashboot\drivers	驱动源文件目录，包括Flash、ADC、EFUSE驱动等。
	boot\flashboot\common	通用组件源文件目录，包括NV接口、分区表接口等。
	boot\flashboot\upg	升级功能源文件目录。
	boot\flashboot\lzma	FlashBoot压缩文件目录。
	boot\flashboot\secure	FlashBoot加密文件目录。
	boot\flashboot\lib	库文件目录。
	Makefile	FlashBoot makefile编译脚本。
	module_config.mk	FlashBoot脚本配置文件。
	SConscript	SCons编译脚本。
loaderboot	boot\loaderboot\fixed\include	芯片固化接口头文件目录。
	boot\loaderboot\include	LoaderBoot头文件目录。
	boot\loaderboot\startup	启动汇编及主程序入口目录。
	boot\loaderboot\drivers	驱动源文件目录，包括Flash、ADC、EFUSE驱动等。
	boot\loaderboot\common	通用组件源文件目录，包括NV接口、分区表接口等。
	boot\loaderboot\secure	LoaderBoot加密文件目录。
	Makefile	LoaderBoot Makefile编译脚本。
	module_config.mk	LoaderBoot脚本配置文件。
	SConscript	SCons编译脚本。



目录名	路径	说明
commonboot	boot \commonboot \crc32	FlashBoot与LoaderBoot公用的CRC32驱动。
	boot \commonboot \efuse	FlashBoot与LoaderBoot公用的EFUSE驱动。
	boot \commonboot \flash	FlashBoot与LoaderBoot公用的Flash驱动。



5 FlashBoot 二次开发指南

5.1 FlashBoot编译

5.2 调整内存布局

5.3 FlashBoot安全启动配置说明

5.4 Boot可用API

5.1 FlashBoot 编译

根目录下执行“sh build.sh”可同时编译Kernel和FlashBoot，FlashBoot编译结果输出为：

- output\bin\Hi3861_boot_signed.bin：写入Flash头部的FlashBoot镜像。
- output\bin\Hi3861_boot_signed_B.bin：写入Flash尾部的FlashBoot备份镜像。

FlashBoot默认使用SHA256签名方式，可直接编译，如果需要使用其他方式签名请参见“[5.3 FlashBoot安全启动配置说明](#)”。

5.2 调整内存布局

FlashBoot的内存布局请参见build\scripts目录下的flashboot_sha256.lds、flashboot_rsa.lds、flashboot_ecc.lds文件，不同后缀的布局用于对应签名方式的FlashBoot。

当用户开发代码后，有可能引起空间不足而链接错误，例如：错误打印如[图5-1](#)所示。

图 5-1 空间不足而链接错误的打印示例

```
Compile /home/wifi/wangjian/proj/1224/code/boot/flashboot/arch/risc-v/hil131h/riscv_init.S
/toolchain/hcc_riscv32_b023/bin/../lib/gcc/riscv32-unknown-elf/7.3.0/../../../../riscv32-unknown-elf/bin/ld: out/hil131_flash_boot.elf section '.text' will not fit in region 'FLASH_BOOT_ADDR'
/toolchain/hcc_riscv32_b023/bin/../lib/gcc/riscv32-unknown-elf/7.3.0/../../../../riscv32-unknown-elf/bin/ld: region 'FLASH_BOOT_ADDR' overflowed by 13072 bytes
collect2: error: ld returned 1 exit status
Makefile:146: recipe for target 'out/hil131_flash_boot.elf' failed
make: *** [out/hil131_flash_boot.elf] Error 1
```

修改方法如下：

步骤1 打开链接文件“build\scripts\flashboot_xxx.lds”。



步骤2 当前内存使用情况如图5-2所示，用户可以根据使用情况调整报错段落的大小和上下相关区域起始地址，图5-2的区域不可重合。

图 5-2 当前内存使用情况示意图

STACK	8KB	0x00100000
SRAM	8KB	0x00102000
ROM_BSS_DATA	2KB	0x00104000
CODE_ROM_BSS_DATA	2KB	0x00104800
HEAP	20KB	0x00105000
SIGN	Sign_len	0x0010A000
FLASH_BOOT	24KB-Sign_len	0x0010A000+Sign_len
CUSTOMER_RSVD	56KB	0x00110000
		0x0011E000

各区域说明如下：

- STACK：运行时的栈空间配置，有栈溢出问题时修改此空间大小。
- SRAM：FlashBoot独有数据段。
- ROM_BSS_DATA：RomBoot与FlashBoot共用数据段，内容不可修改。
- CODE_ROM_BSS_DATA：RomBoot与FlashBoot共用数据段，内容不可修改。
- HEAP：运行时堆空间，用于运行过程中动态申请使用。
- SIGN：FlashBoot签名区，此区域长度与签名方式相关，对应关系为如下：
 - SHA256 签名长度：0x40
 - RSA_V15/RSA_PSS 签名长度：0x5A0
 - ECC签名长度：0x150
- FLASH_BOOT：FlashBoot image加载区，包括FlashBoot签名头总共预留24KB。
- CUSTOMER_RSVD：用户预留区，如STACK、SRAM、HEAP、FLASH_BOOT空间不足可使用此区域。
- FIXED_ROM：RomBoot与FlashBoot共用代码段，内容不可修改。
- CODE_ROM：RomBoot与FlashBoot共用代码段，内容不可修改。

----结束

5.3 FlashBoot 安全启动配置说明

FlashBoot支持三级安全保护，安全性能逐级递增，SDK默认为最低安全级别（SHA256签名），具体如下：

- FlashBoot用SHA256签名，RomBoot通过校验FlashBoot镜像的SHA256值判断其完整性后引导启动FlashBoot。
- FlashBoot用RSA/ECC签名，RomBoot通过EFUSE中的根密钥Hash及FlashBoot的签名数据判断其合法性后引导启动FlashBoot。
- FlashBoot用RSA/ECC签名，并且其代码段用AES-CBC方式加密，RomBoot通过EFUSE中的根密钥盐值生成密钥后配合FlashBoot签名中的IV对FlashBoot代码段进行解密，然后再进行RSA/ECC验签，判断其合法性后引导启动FlashBoot。

说明

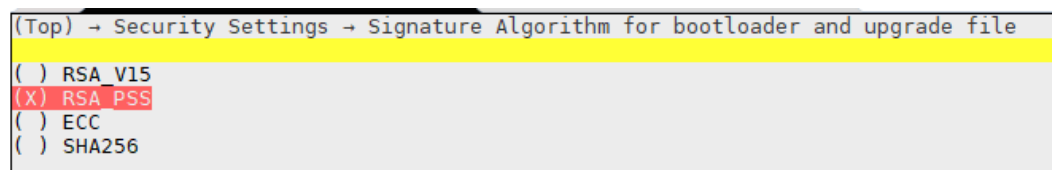
安全启动详细内容请参见《Hi3861V100 / Hi3861LV100 二次开发网络安全 注意事项》。

5.3.1 签名及加密配置

用户可通过Menuconfig->Security Settings 配置FlashBoot签名方式，支持的签名方式有以下几种：

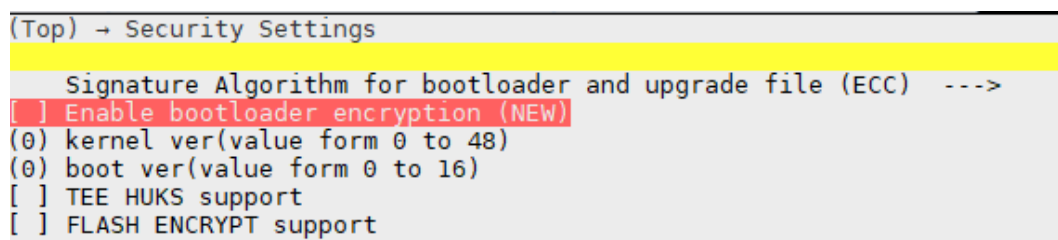
- SHA256：仅计算FlashBoot bin的SHA256值签名。
- RSA_V15：根密钥RSA 4096bit，二级密钥RSA2048bit，使用PKCS1_V15填充方式签名。
- RSA_PSS：根密钥RSA 4096bit，二级密钥RSA2048bit，使用PKCS1_PSS填充方式签名。
- ECC：ECDH_BRAIN_POOL_P256R1密钥签名。

图 5-3 签名方式配置示例



用户可通过Menuconfig配置FlashBoot是否加密，当签名方式选择RSA_V15、RSA_PSS、ECC三种时，Menuconfig会显示下级菜单Enable bootloader encryption，配置此项后签名生成的FlashBoot的代码段会加密。

图 5-4 加密使能配置示例





说明

- Menuconfig配置操作请参见《Hi3861V100 / Hi3861LV100 SDK开发环境搭建 用户指南》。
- SDK包中没有签名和加密需要的密钥，需用户自己生成，密钥要求请参见“[5.3.4 密钥文件说明](#)”。

5.3.2 用户根公钥配置

用户根公钥以明文方式保存在FlashBoot代码的g_boot_rsa_key或g_boot_ecc_key数组中（位置在“boot\flashboot\upg\boot_upg_check_secure.c”）。从用户根密钥证书中读取公钥后将其填入上面数组中即可。

Linux下使用openssl读取公钥可参考以下命令：

```
openssl rsa -in rsa.pem -noout -text
openssl ec -in ecc.pem -noout -text
```

说明

- 读取RSA格式，输出的modulus部分去除第一个byte后是其公钥值。
- 读取ECC格式，输出的pub部分去除第一个byte后是其公钥值。

5.3.3 签名工具介绍

SDK提供了参考的FlashBoot签名工具，路径为“tools/sign_tool/sign_tool”。Linux环境下直接运行此工具可获得帮助说明如[图5-5](#)所示。

图 5-5 sign_tool 工具帮助说明

```
sign_tool version 1.2

sign_tool: [options]
  -h help
  -i [input file path]
  -o [output file]
  -r [root key file]          RSA4096 or ECDSA BRAIN_P00L_P256R1
  -s [sub key file]          RSA2048 or ECDSA BRAIN_P00L_P256R1
  -c [sub key category]      sub key category [0: 0xFFFFFFFF]
  -l [sub key id]            sub key id [0: 23]
  -d [die id]                must be 48 hex nums
  -a [sign alg]              0:RSA PKCS1_V15; 1: RSA PKCS1_PSS; 2: ECC BRAIN_P00L_P256R1
  -v [boot_ver]              range [0, 16], decimal
  -e [aes key file]
  -t generate tail flashboot
  -n Non security bin
  -u [aes key file] encryption upgrade bin
  -f [offset] hexadecimal num,upgrade bin encryption address offset
  -z [length] hexadecimal num,upgrade bin encryption length

non security example:
./sign_tool -i flash_boot.bin -o flash_boot_nos.bin -n

security example:
./sign_tool -i flash_boot.bin -o flash_boot_r.bin -r root_rsa.pem -s sub_rsa.pem -a 1 -e key.txt
```

各项参数含义说明如下：

- -h: 帮助说明。
- -i: 输入文件，需带参数路径及文件名。
- -o: 输出FlashBoot_A文件，需带参数路径及文件名。
- -r: 根密钥，需带参数路径及文件名。

- -s: 二级密钥, 需带参数路径及文件名。
- -e: 加密密钥, 需带参数路径及文件名。
- -t: 输出FlashBoot_B文件, 需带参数路径及文件名。
- -c: 二级密钥类型, 其值为32位无符号整数。
- -l: 二级密钥ID, 其值范围为[0:23]。
- -v: FlashBoot版本号, 其值范围为[0:16]。
- -n: SHA256方式签名。
- -a: 签名算法, 0: RSA_V15; 1: RSA_PSS; 2: ECC。
- -d: 芯片DIE ID, 带此参数签名的FlashBoot只能在对应DIE ID的芯片上使用, 输入值必须为48byte十六进制数据。
用户可使用此工具传入所需参数对FlashBoot进行客制化签名。
- -u: [aes key file]加密升级文件。
- -f: [offset]加密升级的偏移地址, 十六进制数据。
- -z: [length]加密升级的长度, 十六进制数据。

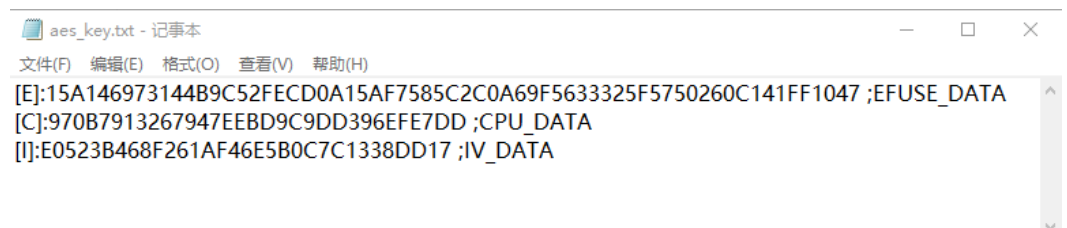
5.3.4 密钥文件说明

- SHA256签名不需要密钥。
- RSA_V15/RSA_PSS签名需要RSA密钥, 其中根密钥为RSA 4096bit, 二级密钥为RSA 2048bit, 将生成的密钥重命名为“root_rsa.pem”、“sub_rsa.pem”, 放于“tools\sign_tool\”目录下供使用, Linux下使用openssl库生成密钥可参考以下命令:

```
openssl genrsa -out root_rsa.pem 4096
openssl genrsa -out sub_rsa.pem 2048
```
- ECC签名需要ECC密钥, 根密钥和二级密钥均须为ECDH_BRAIN_POOL_P256R1格式, 将生成的密钥重命名为“root_ecc.pem”、“sub_ecc.pem”, 放于“tools\sign_tool\”目录下供使用, Linux下使用openssl库生成密钥可参考以下命令:

```
openssl ecparam -genkey -name brainpoolP256r1 -out root_ecc.pem
openssl ecparam -genkey -name brainpoolP256r1 -out sub_ecc.pem
```
- 加密密钥文件需写入三个值:
 - EFUSE_DATA: 32byte, 必须与EFUSE的root_key区写入值一致, 是KDF算法生成密钥所需的HUK。
 - CPU_DATA: 16byte随机数, 用于KDF算法生成密钥的IV前半部分, 会与RomBoot中的另16byte拼成一个完整的32byte值用于KDF生成密钥的IV。
 - IV_DATA: 16byte随机数, AES-CBC加密使用的IV。将写好的密钥文件重命名为“aes_key.txt”, 放于“tools\sign_tool\”目录下供使用, 格式如图5-6所示。

图 5-6 aes_key.txt 文档格式





5.4 Boot 可用 API

Boot API请参见《Hi3861V100 / Hi3861LV100 Boot API 开发参考》。