海思智能机顶盒 1 级安全方案

使用指南

文档版本　　00B04

发布日期　　2016-06-07

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总部　邮编：518129

网址：　http://www.hisilicon.com

客户服务邮箱：　support@hisilicon.com

前 言

概述

本文档主要介绍海思智能机顶盒 1 级安全方案的功能、开发过程中的使用方法以及注意事项。

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
Hi3716C 芯片	V2XX
Hi3716M 芯片	V4XX
Hi3718C 芯片	V1XX
Hi3718M 芯片	V1XX
Hi3719C 芯片	V1XX
Hi3719M 芯片	V1XX
Hi3796C 芯片	V1XX
Hi3798C 芯片	V1XX
Hi3798M 芯片	V1XX
Hi3796M 芯片	V1XX

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
2014-05-26	00B01	第 1 次临时版本发布。
2014-10-31	00B02	新增支持 Hi3796MV100 芯片。
2014-11-25	00B03	将 HISILICON_SECURITY 修改为 HISILICON_SECURITY_L1。 增加服务器配置查看的说明。
2016-06-07	00B04	修改文档名称为使用指南。

目 录

插图目录

1 概述

1.1 背景

顺应市场的安全需求，需要提供 1 级防刷机方案。

1.2 海思智能机顶盒 1 级安全方案的主要功能

海思智能机顶盒 1 级安全方案主要包含如下功能：

- 支持 user 发布模式。
- 防 ROOT 破解。
- 防止 APK 在主系统中替换分区。
- 防止 Launcher 被非法替换。
- 防止非法升级包刷机。
- 提供生成签名公私钥对的命令。
- 编译时，支持开启、关闭本安全方案。

1.3 主要内容

海思智能机顶盒 1 级安全方案中一些功能的使能需要机顶盒厂家的参与，后文主要对相关内容进行介绍：

- 海思智能机顶盒 1 级安全方案适用的芯片范围。
- Android 发布模式的选择。
- Android 的相关配置操作。
- Recovery 镜像和升级包的生成方式。
- PCB 上串口接口的处理。

2 适用芯片

2.1 海思智能机顶盒 1 级安全方案的适用范围

海思智能机顶盒 1 级安全方案适用的芯片范围为：

本安全方案适用于所有普通芯片。包括 Hi3716C、Hi3716M、Hi3718C、Hi3718M、Hi3719C、Hi3719M、Hi3798C、Hi3796C、Hi3798M、Hi3796M 等。

3 Android 发布模式

3.1 Android 发布模式介绍

海思 Android 发布模式可以选择如下 2 种：

- eng 模式
- user 模式

这 2 种发布模式的特性对比如表 3-1 所示。

表3-1 发布模式的特性对比

特性	eng 模式	user 模式
adb 默认使能	YES	NO
adb shell 权限	# root 模式，随便删除或卸载所有应用	$ 没有权限删除系统中的固定软件
串口	打开串口调试	关闭串口调试 （串口不能操作）

3.2 Android 发布模式选择

对比“3.1 Android 发布模式介绍”节中 2 种发布模式的特性，可知，Android user 模式具有更高的安全性。

因此，在编译 Android 时，请使用 user 模式。

4 Android 配置

4.1 概述

海思智能机顶盒 1 级安全方案：

- 可以将分区加载为 read only 模式，防止分区被 APK 进行非法写操作，防止分区被替换或破坏；
- 提供了 Launcher 防替换机制，可以防止 Launcher 被非法替换；
- 提供了 recovery 升级的签名校验机制，对升级包进行校验，防止不合法升级包刷机。

为了支持使能或禁用如上 3 个功能，需要在编译 Android 前进行一些配置操作。本安全方案增加了一个编译开关，能够方便地使能或禁用相关功能，从而简化配置操作。“4.2 编译开关的使用”节将对编译开关的使用进行介绍。

Recovery 升级的签名校验机制，需要使用机顶盒厂家自定义的签名 key，在制作升级包时进行签名，在 recovery 升级时进行签名校验。签名 key 即签名公私钥对。本安全方案提供了生成签名公私钥对的命令，即 mkkey.sh 脚本。“4.3 公私钥对的生成”——“4.5 公私钥对的维护”节将对 mkkey.sh 的使用、公私钥对的使用及维护进行介绍。

4.2 编译开关的使用

为了简化配置操作，增加了一个编译开关 HISILICON_SECURITY_L1，支持开启、关闭本安全方案。

Android 4.2.2 与 Android 4.4.2 中，打开或关闭编译开关时，需要修改的文件有所差异：

对于 Android 4.2.2，在编译 Android 前，修改 device/hisilicon/{CHIPNAME}/device.mk 中“HISILICON_SECURITY_L1”的值（其中“{CHIPNAME}”为 chip 的名字，例如，Hi3716CV200 芯片需要修改 device/hisilicon/Hi3716CV200/device.mk）；

对于 Android 4.4.2，在编译 Android 前，修改 device/hisilicon/{CHIPNAME}/customer.mk 中“HISILICON_SECURITY_L1”的值。

“HISILICON_SECURITY_L1”的修改方法如下：

- 开启本安全方案：
  HISILICON_SECURITY_L1 := true
- 关闭本安全方案：
  HISILICON_SECURITY_L1 := false

4.3 公私钥对的生成

为了防止非法升级包刷机，在 recovery 升级程序中增加签名校验机制，对升级包进行签名校验。签名和校验时需要使用机顶盒厂家自定义的签名 key。制作升级包时使用公钥和私钥进行签名，recovery 升级时使用公钥进行签名校验，合法则成功升级，非法则刷机失败。

提供了 mkkey.sh，用于生成公私钥对，并存放到指定目录。

mkkey.sh 脚本的使用方法如下：

在编译 Android 前，进入 Android 代码的根目录，顺序执行如下操作（以 Hi3716CV200 芯片为例）：

步骤 1 切换为 root 用户；

步骤 2 键入命令：source build/envsetup.sh

步骤 3 键入命令：lunch Hi3716CV200-user

步骤 4 键入命令：sh device/hisilicon/bigfish/build/mkkey.sh <输入参数>

步骤 5 退出 root。

注：如果通过“which mknod | xargs ls –l”、“which mktemp | xargs ls -l”、“which openssl | xargs ls -l”、“which tee | xargs ls -l”等命令查看服务器配置，发现 mknod、mktemp、openssl、tee 等的权限都是 0755，user 和 group 都是 root 时，步骤 1 和步骤 5 可以跳过。例如：

```
user@server:~$ which mknod | xargs ls -l
-rwxr-xr-x 1 root root 31168 Nov 20  2012 /bin/mknod
```

步骤 4 中，“<输入参数>”为厂商自身的认证信息（Authentication Information），如果未输入参数，系统将默认使用“'/C=CN/ST=Guangdong/L=Shenzhen/O=Android/OU=Android/CN=Android/emailAddress=android@android.com'”。

该“<输入参数>”需要遵循如下格式：

'/C=<country>/ST=<province>/L=<city>/O=<organization>/OU=<organization Unit>/CN=<common name>/emailAddress=<email>'，

其中：

- “<country>”为 Country Code，由 2 个英文大写字母表示。例如，美国为 US。
- “<province>”为 State or Province Name。例如，California。
- “<city>”为 Locality or City Name。例如，MountainView。

- “<organization>”为 Organization Name。例如，Android。
- “<organization Unit>”为 Organizational Unit Name。例如，Android。
- “<common name>”为 Common Name，即 your name or your server’s hostname。
- “<email>”为 Contact Email Address。例如，android@android.com。

如果输入参数不符合上述格式要求，那么生成的 key 中认证信息将丢失或者不完整。请务必遵循上述格式要求。

在使用 mkkey.sh 脚本时，需要注意：

执行 mkkey.sh 脚本的目的是生成 releasekey、platform、shared、media 等 4 对 key，这些 key 将会在系统升级、应用程序更新等功能中用于签名验证。因此：

- key 不能多次生成。

  在使用“sh device/hisilicon/bigfish/build/mkkey.sh <输入参数>”时，如果 key 已经存在了（生成的 key 位于：device/hisilicon/{CHIPNAME}/security/，其中“{CHIPNAME}”为 chip 的名字，例如，Hi3716CV200 芯片的 key 位于 device/hisilicon/Hi3716CV200/security/），会提示如下信息：

  ```
  **.pk8 and/or **.x509.pem already exist; please delete them first
  if you want to replace them.
  ```

  建议只调用一次命令！
- 不要输入密码。

  在“sh device/hisilicon/bigfish/build/mkkey.sh <输入参数>”执行过程中，会出现 4 次提示输入密码的信息：

  ```
  Enter password for '** ' (blank for none; password will be visible):
  ```

  如果此处输入了密码，在 Android 编译过程中或单独进行 apk 签名时，将需要输入此密码。在编译 Android 时为了加速一般会在执行 make 时使用“-j**”（即采用多线程），这样本来需要手动输入密码时，由于其他线程的运行影响当前的输入终端，所以就会导致密码无法输入，报 java.lang.NullPointerException 的错误。因此，请直接敲回车，跳过密码输入。

4.4 公私钥对的使用

mkkey.sh 脚本生成的公私钥为 releasekey、platform、shared、media 等 4 对 key。这 4 对 key 的作用为：

- platform – 用于 core platform 中部分 packages 的签名（a key for packages that are part of the core platform）。
- shared – 用于 home/contacts 进程中共享部分的签名（a key for things that are shared in the home/contacts process）。
- media – 用于 media/download 系统中部分 packages 的签名（a key for packages that are part of the media/download system）。
- releasekey – 用于系统中其他 packages 的签名、升级包（update.zip）的签名和校验。

执行 mkkey.sh 脚本生成的这 4 对公私钥位于 device/hisilicon/{CHIPNAME}/security 目录下（其中“{CHIPNAME}”为 chip 的名字，例如，Hi3716CV200 芯片的 key 位于 device/hisilicon/Hi3716CV200/security/）。

注意

本安全方案的编译系统，能实现自动进行应用程序、升级包等的签名，在 recovery 中增加签名校验功能等。为了使得编译能正常执行，在编译系统中对这些功能所用的 key 已经做出了限定。因此，在编译 Android 前，请勿更改这 4 对 key 的存放位置，请勿替换或修改其中的任何一个 key。

4.5 公私钥对的维护

对 mkkey.sh 脚本生成的公私钥对进行维护时，需要注意：

- 请务必保存好并备份这 4 对公私钥

  mkkey.sh 生成的这 4 对公私钥包含厂商自身的认证信息，在系统升级、应用程序更新等功能中将用于签名验证。如果使用签名不一致的升级包尝试系统升级，将会升级失败；如果使用签名不一致的应用程序尝试程序更新操作，更新将会失败。
- 请务必保管好这 4 对公私钥，防止 key 外泄。

  如果 key 已泄露，本安全方案将失效，将无法避免非法升级刷机。

4.6 注意事项

生成、使用及维护公私钥对时，需要注意以下几点：

- 执行 mkkey.sh 脚本时，请遵循相应的格式要求。
- key 不能多次生成。

  建议只运行一次 mkkey.sh 脚本。
- 运行 mkkey.sh 脚本时不要输入密码。

  请直接敲回车，跳过密码输入。
- 请勿更改 key 的存放位置，请勿替换或修改其中的任何一个 key。
- 请务必备份并保管好生成的 key。

5 Android 编译

5.1 概述

完成上一章的 Android 配置后，便可以通过 Android 编译生成：

- 包含签名校验机制的 recovery 镜像；
- 机顶盒厂家自定义 key 签名的升级包（update.zip）。

Recovery 和 update.zip 的生成可以通过完整编译 Android 来实现，也可以单独编译 recovery 和 update.zip。

Android 编译的系统介绍可以参考海思 Android 源代码根目录下的 install_notes_cn.txt 或 install_notes_en.txt。

5.2 完整编译 Android

进入 Android 代码的根目录，顺序执行如下操作（以 Hi3716CV200 芯片为例）：

步骤 1 键入命令：source build/envsetup.sh

步骤 2 键入命令：lunch Hi3716CV200-user

步骤 3 键入命令：make bigfish -j32 2>&1 | tee bigfish.log

编译结果：

在 out/target/product/Hi3716CV200 目录会有 Nand、Emmc 两个目录，其中，Nand 目录下镜像适合烧写到（Spi+Nand）单板，Emmc 目录下镜像适合烧写到 eMMC 单板。

编译生成的 Recovery 镜像和升级包在 Nand 和 Emmc 两个目录下都有：

- Recovery 小系统镜像：
  recovery.img
- 升级包：
  update.zip

5.3 编译 recovery

进入 Android 代码的根目录，顺序执行如下操作（以 Hi3716CV200 芯片为例）：

步骤 1 键入命令：source build/envsetup.sh

步骤 2 键入命令：lunch Hi3716CV200-user

步骤 3 键入命令：make recoveryimg -j32 2>&1 | tee recovery.log

编译结果：

在 out/target/product/Hi3716CV200 目录的 Nand、Emmc 两个目录下都会生成 recovery.img。两份 recovery.img 是一样，只是分别拷贝到了 Nand、Emmc 目录。

注意

Recovery 小系统的编译命令可以直接执行，不需要完整编译 Android 以后再执行。

5.4 编译 update.zip

进入 Android 代码的根目录，顺序执行如下操作（以 Hi3716CV200 芯片为例）：

步骤 1 键入命令：source build/envsetup.sh

步骤 2 键入命令：lunch Hi3716CV200-user

步骤 3 键入命令：make updatezip -j32 2>&1 | tee updatezip.log

编译结果：

在 out/target/product/Hi3716CV200 目录的 Nand、Emmc 两个目录下都会生成 update.zip。

注意

生成 update.zip 的编译命令可以直接执行，不需要完整编译 Android 以后再执行。

6 PCB 的处理

6.1 PCB 上串口处理的目的

对 PCB 上的串口接口进行处理的目的：

通过设置一定的硬件门槛，防止普通刷机者通过串口进行刷机。

6.2 PCB 上串口处理的方式

为了提高安全性能，需要对 PCB 做如下处理：

设备出厂之后，机顶盒厂家在 PCB 上不保留串口接口，不焊串口接口排针、不提供标准串口转换器等。

PCB 不焊接串口排针，可以设置一定的硬件门槛，于是，开发维修人员与普通刷机者的操作便有所差异，如图 6-1 所示。

图6-1 硬件门槛示意图