Level-1 Security Solution for the HiSilicon Intelligent STB

User Guide

Issue	**00B04**
Date	**2016-06-07**

Trademarks and Permissions

Notice

The purchased products, services and features are stipulated by the contract made between HiSilicon and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

HiSilicon Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: http://www.hisilicon.com

Email: support@hisilicon.com

About This Document

Overview

This document describes the functions and usage of the level-1 security solution for the HiSilicon intelligent set top box (STB), as well as the precautions to be taken when the STB is used.

Product Version

The following table lists the product versions related to this document.

Product Name	Product Version
Hi3716C	V2XX
Hi3716M	V4XX
Hi3718C	V1XX
Hi3718M	V1XX
Hi3719C	V1XX
Hi3719M	V1XX
Hi3796C	V1XX
Hi3798C	V1XX
Hi3798M	V1XX
Hi3796M	V1XX

Intended Audience

This document is intended for:

- Technical support engineers
- Software development engineers

Change History

Changes between document issues are cumulative. Therefore, the latest document issue contains all changes made in previous issues.

Issue 00B04 (2016-06-07)

This issue is the fourth draft release, which incorporates the following change:

The name of this document is changed.

Issue 00B03 (2014-11-25)

This issue is the third draft release, which incorporates the following change:

HISILICON_SECURITY is changed to HISILICON_SECURITY_L1.

Issue 00B02 (2014-10-31)

This issue is the second draft release, which incorporates the following change:

Hi3796M V100 is supported.

Issue 00B01 (2014-05-26)

This issue is the first draft release.

Contents

Figures

Tables

1 Overview

1.1 Background

A level-1 anti-flashing solution must be provided to meet the market security requirements.

1.2 Functions

The level-1 security solution for the HiSilicon intelligent STB has the following functions:

- Supports the user release mode.
- Protects roots from being cracked.
- Prevents partition replacement by application package files (APKs) in the system.
- Protects the launcher from being illegally replaced.
- Prevents ROM flashing by illegal upgrade packages.
- Provides commands to generate signed public and private keys pairs.
- Supports enabling and disabling the security solution during compilation.

1.3 Main Contents

Some functions provided by the level-1 security solution for the HiSilicon intelligent STB must be enabled by assistance of STB vendors. This document describes the following contents:

- Chipsets applicable to the level-1 security solution for the HiSilicon intelligent STB
- Android release modes
- Android-related configurations and operations
- Methods for generating recovery images and upgrade packages
- Methods for processing serial ports on the printed circuit board (PCB).

2 Applicable Chipsets

The level-1 security solution for the HiSilicon intelligent STB applies to the following chipsets:

All common chipsets, including the Hi3716C, Hi3716M, Hi3718C, Hi3718M, Hi3719C, Hi3719M, Hi3796M, Hi3798M, Hi3796C, Hi3798C.

3 Android Release Modes

3.1 Introduction

HiSilicon Android systems can be released in the following two modes:

- eng mode
- user mode

Table 3-1 describes features of the two release modes.

Table 3-1 Features of the two release modes

Feature	eng Mode	user Mode
Default Android debug bridge (ADB) enabling	Yes	No
ADB shell permission	# The root user can delete or uninstall all applications.	$ Users do not have the permission to delete fixed software in the system.
Serial port	Enabling serial port commissioning	Disabling serial port commissioning No operation can be performed on the serial port.

3.2 Android Release Mode Selection

Comparison of the two release modes described in section 3.1 "Introduction" indicates that the user mode has a higher security.

You are advised to use the user mode during Android compilation.

4 Android Configuration

4.1 Overview

The level-1 security solution for the HiSilicon intelligent STB has the following functions:

- Partitions can be loaded in read only mode to protect the partitions from being replaced or damaged and prevent illegal write operations by APKs.
- The launcher anti-replacement mechanism is provided to protect the launcher from being illegally replaced.
- During recovery upgrade, the signature verification mechanism is provided for upgrade packages to prevent ROM flashing by illegal upgrade packages.

To enable or disable the preceding three functions, configuration operations must be performed before Android compilation. A compilation switch is added to this security solution to enable or disable related functions and simplify configuration operations. For details, see section 4.2 "Using the Compilation Switch."

The signature key provided by the STB vendor is required for the signature verification mechanism during recovery upgrade. A signature is created during upgrade package preparation and is verified during recovery upgrade. A signature key is a signed public and private key pair. This solution provides the **mkkey.sh** script to generate signed public and private key pairs. For details about how to use the **mkkey.sh** script and how to use and maintain public and private key pairs, see sections 4.3 "Generating Public and Private Key Pairs" to 4.5 "Maintaining Public and Private Key Pairs."

4.2 Using the Compilation Switch

To simplify configuration operations, compilation switch HISILICON_SECURITY_L1 is added to enable and disable the security solution.

When the compilation switch is turned on or turned off, files to be modified are different in Android 4.2.2 and Android 4.4.2.

For Android 4.2.2, change the value of HISILICON_SECURITY_L1 in the **device/hisilicon/{CHIPNAME}/device.mk** file before compiling the Android system. **{CHIPNAME}** indicates the chip name. For example, change the value of HISILICON_SECURITY_L1 in the **device/hisilicon/Hi3716CV200/device.mk** file for Hi3716C V200.

For Android 4.4.2, change the value of HISILICON_SECURITY_L1 in the **device/hisilicon/Hi3798MV100/customer.mk** file before compiling the Android system.

Set the value of HISILICON_SECURITY as follows to enable or disable the security solution:

- To enable the security solution:

  HISILICON_SECURITY_L1 = true
- To disable the security solution:

  HISILICON_SECURITY_L1= false

4.3 Generating Public and Private Key Pairs

To prevent ROM flashing by illegal upgrade packages, the signature verification mechanism is added in the recovery upgrade program to verify the signature of the upgrade packages. The signature key provided by the STB vendor is required during signature creation and verification. When preparing an upgrade package, use the public and private keys for signature. During a recovery upgrade, use the public key for signature verification. If the upgrade package is legal, the upgrade will be successful. If the upgrade package is illegal, the flashing will fail.

The **mkkey.sh** script is provided to generate public and private key pairs and save the key pairs to the specified directory.

To use the **mkkey.sh** script, perform the following steps:

Before compiling the Android system, enter the root directory of the Android code and perform the following operations in sequence (using Hi3716C V200 as an example):

Step 1 Log in to the system as the **root** user.

Step 2 Run the following command:

```
source build/envsetup.sh
```

Step 3 Run the following command:

```
lunch Hi3716CV200-user
```

Step 4 Run the following command:

```
sh device/hisilicon/bigfish/build/mkkey.sh <input parameter>
```

Step 5 Exit the root.

In Step 4, **<input parameter>** is the authentication information of the vendor. If this parameter is not set, the default value **'/C=CN/ST=Guangdong/L=Shenzhen/O=Android/OU=Android/CN=Android/emailAddress=android@android.com'** is used.

<input parameter> must be set in the following format:

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

where,

- **<country>** indicates the country code, expressed by two uppercase letters. For example, **US** indicates the United States.
- **<province>** indicates the state or province name, for example, **California**.
- **<city>** indicates the locality or city name, for example, **MountainView**.
- **<organization>** indicates the organization name, for example, **Android**.
- **<organization Unit>** indicates the organization unit name, for example, **Android**.
- **<common name>** indicates the user name or user server's host name.
- **<email>** indicates the contact email address, for example, **android@android.com**.

If **<input parameter>** does not meet the preceding format requirements, the authentication information in the generated key will be lost or incomplete. Therefore, make sure that **<input parameter>** is set in compliance with the format requirements.

When using the mkkey.sh script, pay attention to the following items:

Execute the **mkkey.sh** script to generate the **releasekey**, **platform**, **shared**, and **media** key pairs, which are used for signature verification during the system upgrade and application update. Therefore:

- Do not generate the key multiple times.

  If a key exists when the **sh device/hisilicon/bigfish/build/mkkey.sh <input parameter>** command is executed, the following information will be displayed: (The generated key is stored in the **device/hisilicon/{CHIPNAME}/security/** directory. **{CHIPNAME}** indicates the chip name, for example, the key of Hi3716C V200 is stored in the **device/hisilicon/Hi3716CV200/security/** directory.)

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

  You are advised to invoke the command once.
- Do not input the password.

  When the **sh device/hisilicon/bigfish/build/mkkey.sh <input parameter>** command is executed, the following message will be displayed four times, prompting you to input the password:

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

  If a password is set, it must be input during Android compilation or independent APK signing. To accelerate Android compilation, the **make:-j**** command is executed in multi-thread mode. During this process, a password needs to be input manually. However, the password cannot be input because the input terminal is affected by other threads, and the **java.lang.NullPointerException** message indicating an error is displayed. Therefore, press **Enter** to skip password input.

----End

4.4 Using Public and Private Key Pairs

When you execute the **mkkey.sh** script, the **releasekey**, **platform**, **shared**, and **media key** pairs are generated, which have the following functions:

- The **platform** key pair is used for signatures of some packages in the core platform.
- The **shared** key pair is used for signatures of the shared part in the home/contacts process.

- The **media** key pair is used for signatures of some packages in the media/download system.
- The **releasekey** key pair is used for signature creation and verification of other packages and the **update.zip** file.

The four key pairs are stored in the **device/hisilicon/{CHIPNAME}/security/** directory. **{CHIPNAME}** indicates the chip name, for example, the key of Hi3716C V200 is stored in the **device/hisilicon/Hi3716CV200/security/** directory.

⚠ **CAUTION**

The compilation system in the security solution can automatically create signatures for applications and upgrade packages and verify signatures in the recovery program. To ensure proper execution of compilation functions, the keys used for the functions are restricted. Therefore, do not change the storage location of the four key pairs, or replace or modify any of the keys before Android compilation.

4.5 Maintaining Public and Private Key Pairs

When maintaining public and private key pairs generated by executing the **mkkey.sh** script, pay attention to the following items:

- Properly store and backup the four public and private key pairs.

  The four public and private key pairs generated by executing the **mkkey.sh** script include the authentication information of the vendor, which is used for signature verification during system upgrade and application update. If an upgrade package or update application with a different signature is used, the upgrade or update will fail.
- Protect the four public and private key pairs from divulgence.

  If the keys are divulged, the security solution is invalid, and illegal upgrade and ROM flashing may occur.

4.6 Precautions

Pay attention to the following items when generating, using, and maintaining public and private key pairs:

- Execute the mkkey.sh script in compliance with the related format requirements.
- Do not generate the key multiple times.

  You are advised to execute the **mkkey.sh** script once.
- Do not input the password when executing the **mkkey.sh** script.

  Press **Enter** to skip password input.
- Do not change the storage location of the four key pairs, or replace or modify any of the keys.
- Properly back up and store the generated keys.

5 Android Compilation

5.1 Overview

After Android configuration is completed, the Android system can be compiled to generate the following items:

- The recovery image including the signature verification mechanism
- The **update.zip** file whose key signature is provided by the STB vendor

The recovery image and **update.zip** file can be compiled separately, or generated by complete Android compilation.

For details about the Android compilation system, see the **install_notes_cn.txt** or **install_notes_en.txt** file in the root directory of the HiSilicon Android source code.

5.2 Completely Compiling the Android System

Enter the root directory of the Android code and perform the following operations in sequence (using Hi3716C V200 as an example):

Step 1 Run the following command:

```
source build/envsetup.sh
```

Step 2 Run the following command:

```
lunch Hi3716CV200-user
```

Step 3 Run the following command:

```
make bigfish -j32 2>&1 | tee bigfish.log
```

Compilation results:

There are two directories (**Nand** and **Emmc**) in the **out/target/product/Hi3716CV200** directory. You are advised to burn the images in the **Nand** directory to the SPI+NAND board, and to burn the images in the **Emmc** directory to the eMMC board.

The generated recovery image and upgrade package are stored in both the **Nand** and **Emmc** directories.

- Recovery small system image:
  recovery.img
- Upgrade package:
  update.zip

----End

5.3 Compiling the Recovery Image

Enter the root directory of the Android code and perform the following operations in sequence (using Hi3716C V200 as an example):

Step 1 Run the following command:

```
source build/envsetup.sh
```

Step 2 Run the following command:

```
lunch Hi3716CV200-user
```

Step 3 Run the following command:

```
make recoveryimg -j32 2>&1 | tee recovery.log
```

Compilation results:

recovery.img is generated in both the **Nand** and **Emmc** directories in the **out/target/product/Hi3716CV200** directory. The two **recovery.img** files are the same and copied to the **Nand** and **Emmc** directories.

NOTE

The compilation command in the recovery small system can be directly executed, not necessarily after the Android system is completely compiled.

----End

5.4 Compiling the update.zip File

Enter the root directory of the Android code and perform the following operations in sequence (using Hi3716C V200 as an example):

Step 1 Run the following command:

```
source build/envsetup.sh
```

Step 2 Run the following command:

```
lunch Hi3716CV200-user
```

Step 3 Run the following command:

```
make updatezip -j32 2>&1 | tee updatezip.log
```

Compilation results:

update.zip is generated in both the **Nand** and **Emmc** directories in the **out/target/product/Hi3716CV200** directory.

NOTE

The compilation command for generating the **update.zip** file can be directly executed, not necessarily after the Android system is completely compiled.

----End

6 PCB Processing

6.1 Purpose for Processing Serial Ports on the PCB

After processing serial ports on the PCB, hardware thresholds are set to prevent common ROM flashing or ROM flashing over serial ports.

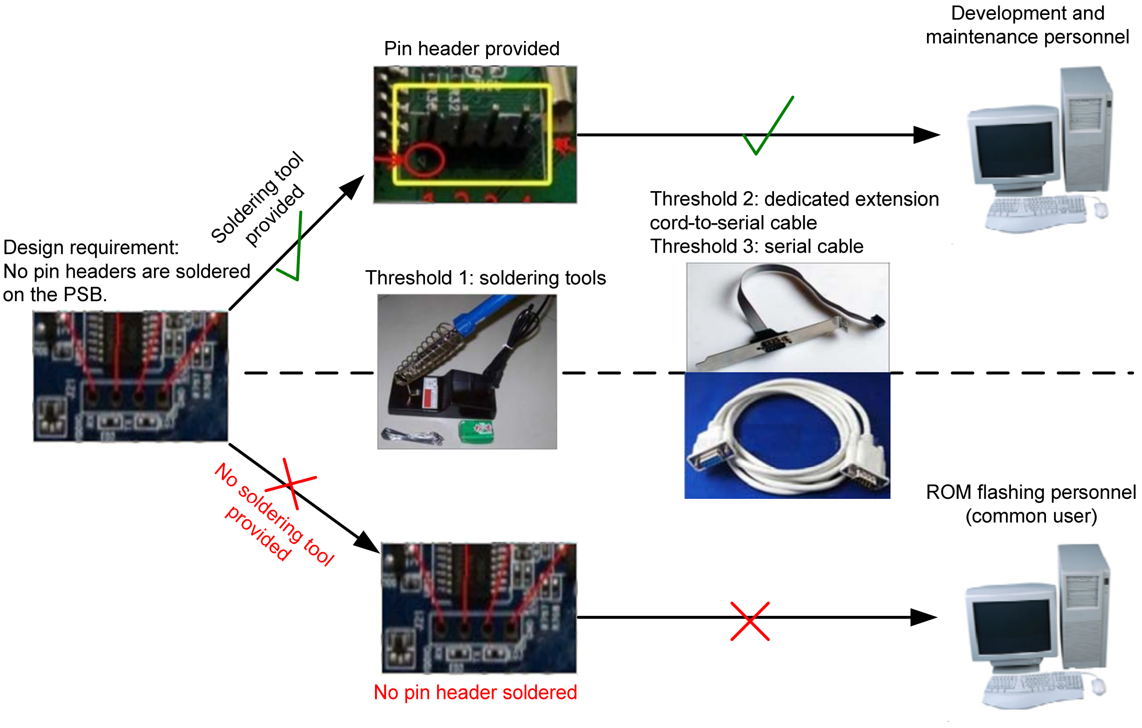
6.2 Methods for Processing Serial Ports on the PCB

To improve the security performance, perform the following operations for the PCB:

Before a PCB is delivered, the STB vendor cannot reserve serial ports on the PCB, solder pin headers on the serial ports, or provide standard serial port converters.

Set hardware thresholds to restrict pin header soldering on the PCB. Figure 6-1 shows different operations performed by development and maintenance personnel and common ROM flashing personnel.

Figure 6-1 Setting hardware thresholds


Development and maintenance personnel
Pin header provided
Soldering tool provided
Design requirement:
No pin headers are soldered on the PSB.
Threshold 2: dedicated extension cord-to-serial cable
Threshold 3: serial cable
Threshold 1: soldering tools
No soldering tool provided
ROM flashing personnel (common user)
No pin header soldered