

# Rockchip RK3588 HDCP 开发指南

文件标识：RK-KF-YF-768

发布版本：V1.0.0

日期：2023-07-21

文件密级：绝密 秘密 内部资料 公开

## 免责声明

本文档按“现状”提供，瑞芯微电子股份有限公司（“本公司”，下同）不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因，本文档将可能在未经任何通知的情况下，不定期进行更新或修改。

## 商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标，归本公司所有。

本文档可能提及的其他所有注册商标或商标，由其各自拥有者所有。

## 版权所有 © 2023 瑞芯微电子股份有限公司

超越合理使用范畴，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。

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址：福建省福州市铜盘路软件园A区18号

网址：[www.rock-chips.com](http://www.rock-chips.com)

客户服务电话：+86-4007-700-590

客户服务传真：+86-591-83951833

客户服务邮箱：[fae@rock-chips.com](mailto:fae@rock-chips.com)

## 前言

本文主要介绍 Rockchip RK3588 平台 HDMI TX / HDMI RX / DPTX 的 HDCP1.4 和 HDCP2.3 配置说明和常见问题分析。

## 产品版本

芯片名称	Kernel版本	Android版本
RK3588	Linux 5.10	Android12、Android13

## 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

## 修订记录

版本号	作者	修改日期	修改说明
V1.0.0	陈顺庆、操瑞杰、张玉炳	2023-07-21	初始发

## 目录

### Rockchip RK3588 HDCP 开发指南

HDCP 概述

HDCP 工具说明

HDCP 支持不同的烧写模式

HDCP 两种不同烧写模式的区别

加密烧写库 libhdcp.so 接口

HDCP 1.4 KEY 的加密说明

HDCP 2.3 固件打包说明

Firmware 和 RX KEY 加密 (可选项, 非必须)

HDCP1.4 配置及说明

HDMIRX

dts 配置

Key 烧写

RK 工具烧写

客户自行开发工具烧写

HDMITX

DPTX

dts 配置

Key 烧写

HDCP2.3 配置及说明

HDMIRX

dts 配置

打包 Firmware 和 启动服务

HDMITX

dts 配置

打包 Firmware 和 启动服务

DPTX

dts 配置

打包 Firmware 和 启动服务

HDCP 状态查看

HDMIRX

HDMITX

DPTX

常见问题分析

hdcp2\_tx\_rx 启动失败

HDMIRX 黑屏或是绿屏插拔无法恢复

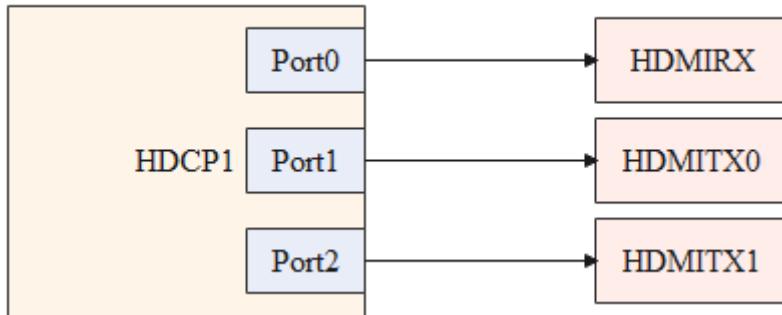
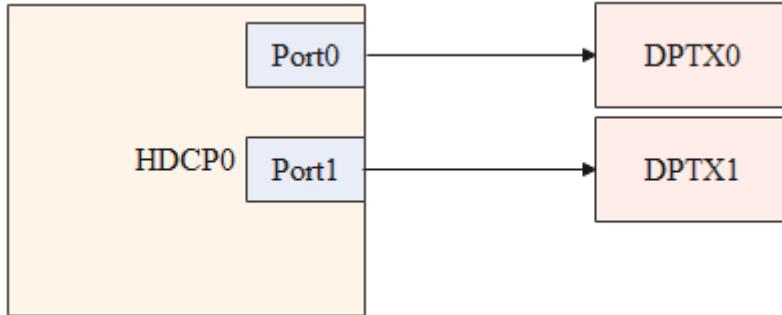
同时打开 HDMITX + HDMIRX 的 HDCP2.3, 拔出 RX 的时候 TX 会闪一下花屏

## HDCP 概述

- RK3588 平台, HDMITX、HDMIRX、DPTX 分别支持 HDCP1.4 和 HDCP2.3.
- RK3588 平台有两个独立的 HDCP2.3 控制器 (hdcp0 和 hdcp1), 每个 HDCP2.3 控制器都有 3 个 Port:

hdcp0: DPTX0 和 DPTX1

hdcp1: HDMIIRX、HDMITX0 和 HDMITX1



- HDCP1.4 或是 HDCP2.3 KEY 都需要客户自行到 HDCP 官网购买，RK3588 没有内置 HDCP KEY.

## HDCP 工具说明

### HDCP 支持不同的烧写模式

1. RK 提供对应的 KEY 转换工具和烧写工具，可以从 SDK 从获取：

Key 拆分和转换工具：

RKTools/windows/Rockchip\_HdcpKey\_Writer\_v1.0.1.7z 工具包下面有 KeyConvertor key 的拆分和转换工具，更新到 KeyConvertor-v2.0.5 可以拆分成一个bin文件只有一个key.

Key 烧写工具：

RKTools/windows/Rockchip\_HdcpKey\_Writer\_v1.0.1.7z

2. RK 提供加密烧写库 libhdcp.so，客户可以根据自己的需求开发对应的烧写工具，对应的 libhdcp.so 需要提 redmine 获取。

### HDCP 两种不同烧写模式的区别

1. RK 提供的烧写工具，在 KEY 的 SEED 加密基础上会再进行 AES 加密，然后写入 Vendor Storage。
2. 加密烧写库 libhdcp.so 只支持 SEED 加密。

### 加密烧写库 libhdcp.so 接口

```
enum HDCP_KEY_ID {
    HDCP1X_KEY_HDMITX_RK33 = 0,
    HDCP1X_KEY_HDMITX_RK3588,
    HDCP1X_KEY_HDMIIRX_RK3588,
```

```

HDCP1X_KEY_HDMIRX_RK628,
HDCP1X_KEY_DP_RK3588,
HDCP2X_KEY_HDMIRX_RK3588,
HDCP2X_KEY_RKIV_RK3588,
};

/*
 * Encrypt the key, but is not written to vendor
 *
 * id: HDCP KEY type
 * keyin: HDCP1.x 308 Byte raw Key, HDCP2.x 1000 Byte Key
 * keyin_size: HDCP1.x 308, HDCP2.x 1000
 * keyout: The encrypted key
 * keyout_size: HDCP1.x 314, HDCP2.x 1000
 */
int hdcp_key_process(enum HDCP_KEY_ID id, uint8_t *keyin, int keyin_size,
uint8_t *keyout, int keyout_size);

/*
 * Encrypt key and write it to vendor
 *
 * id: HDCP KEY type
 * keyin: HDCP1.x 308 Byte raw Key, HDCP2.x 1000 Byte Key
 * keyin_size: HDCP1.x 308, HDCP2.x 1000
 */
int hdcp_key_process_and_write(enum HDCP_KEY_ID id, uint8_t *keyin, int
keyin_size);

/*
 * Write key to vendor
 *
 * id: HDCP KEY type
 * key: HDCP1.x 308 Byte raw Key, HDCP2.x 1000 Byte Key
 * size: key size
 */
int hdcp_key_write(enum HDCP_KEY_ID id, uint8_t *key, int size);

/*
 * Read key from vendor
 *
 * id: HDCP KEY type
 * key: HDCP1.x 314 Byte Key, HDCP2.x 1000 Byte Key
 * len: key length
 */
int hdcp_key_read(enum HDCP_KEY_ID id, uint8_t *key, int *len);

/*
 * Write rkiv to vendor
 *
 * rkiv: rkiv
 * len: must be 16
 */
int hdcp_rkiv_write(uint8_t *rkiv, int len);

```

## HDCP 1.4 KEY 的加密说明

一个HDCP 1.4 KEY有308 Bytes，分别如下：  
8 Bytes KSV: 5 Bytes KSV + 3 Bytes 0x0组成；  
280 Bytes DPK；  
20 Bytes SHA。

SEED 加密，只对 280 Bytes 的 DPK进行加密。

## HDCP 2.3 固件打包说明

DWC\_HDCP2\_BASE\_ESM\_Firmware 固件打包工具没有随 SDK 发布，所以需要 redmine 或是邮件向 RK 获取。

### 1. 单独 HDMIRX 固件和 KEY

RX KEY 放到打包工具的根目录和 tools目录下：

```
DWC_HDCP2_BASE_ESM_Firmware$  
cp  ../hdcp_receivers.bin .  
cp  ../hdcp_receivers.bin tools/
```

执行**build\_rockchip\_fw.sh** 脚本，选择 1 -build firmware for both HDMIRX and HDMIITX，生成 **./firmware/hdpc2\_hdmi.fw** 固件：

```
1 -build firmware for both HDMIRX and HDMIITX  
2 -build firmware for HDMIITX only  
3 -build firmware for DP  
Choose the fw type: 1
```

### 2. HDMIRX + HDMIITX 固件和 KEY

RX + TX KEY 放到打包工具的根目录和 tools目录下：

```
DWC_HDCP2_BASE_ESM_Firmware$  
cp  ../hdcp_receivers.bin .  
cp  ../hdcp_receivers.bin tools/  
cp  ../hdcp_transmitter.bin .  
cp  ../hdcp_transmitter.bin tools/
```

执行**build\_rockchip\_fw.sh** 脚本，选择 1 -build firmware for both HDMIRX and HDMIITX，生成 **./firmware/hdpc2\_dp.fw** 固件：

```
1 -build firmware for both HDMIRX and HDMIITX  
2 -build firmware for HDMIITX only  
3 -build firmware for DP  
Choose the fw type: 1
```

### 3. 单独 HDMIITX 固件

TX KEY 放到打包工具的根目录和 tools目录下：

```
DWC_HDCP2_BASE_ESM_Firmware$  
cp  ../hdcp_transmitter.bin .  
cp  ../hdcp_transmitter.bin tools/
```

执行**build\_rockchip\_fw.sh** 脚本, 选择 2 -build firmware for HDMITX only, 生成  
**./firmware/hdpc2\_hdmi.fw** 固件:

```
1 -build firmware for both HDMIRX and HDMITX
2 -build firmware for HDMITX only
3 -build firmware for DP
Choose the fw type: 2
```

#### 4. DPTX 固件

TX KEY 放到打包工具的根目录和 tools 目录下:

```
DWC_HDCP2_BASE_ESM_Firmware$ 
cp  ../hdcp_transmitter.bin .
cp  ../hdcp_transmitter.bin tools/
```

执行**build\_rockchip\_fw.sh** 脚本, 选择 3 -build firmware for DP, 生成  
**./firmware/hdpc2\_hdmi.fw** 固件:

```
1 -build firmware for both HDMIRX and HDMITX
2 -build firmware for HDMITX only
3 -build firmware for DP
Choose the fw type: 3
```

## Firmware 和 RX KEY 加密 (可选项, 非必须)

Firmware 和 RX KEY 默认会有 PKf / DUK 加密, 然后在 HDCP 控制器 load Firmware 和 RX KEY 之后用 PKf / DUK 进行解密, PKf / DUK 在芯片出厂的时候烧录到 OTP。

为了确保每个客户 Firmware 和 RX KEY 的唯一性, 可以对 Firmware 和 RX KEY 在打包的时候用密钥 RKIV 多一层 AES 加密, 然后 HDCP2 服务 load Firmware 和 RX KEY 之后会先进行 AES 解密, 然后再传给 HDCP 控制器进行 Kf / DUK 解密, 具体配置如下:

1. firmware.aic.nouart 配置 16 bytes 的 RKIV, RKIV 为加密密钥, 可自行定义, 但长度必须是 16 bytes:

```
--- a/DWC_HDCP2_BASE_ESM_Firmware/firmware/firmware.aic.nouart
+++ b/DWC_HDCP2_BASE_ESM_Firmware/firmware/firmware.aic.nouart
@@ -2,6 +2,7 @@ PKf      = 0x00112233445566778899aabbccddeeff
 DUK      = 0xffeeddccbbaa99887766554433221100
 IK       = 0xffeeddccbbaa99887766554433221100
 IVC      = Oxdeadbeeffedcba9876543210
+RKIV     = 0x00112233445566778899aabbccddeeff
 BBRcode  = 6144
 BBRdwb   = 3072
 CLL      = 64
```

2. 调用 libhdcp.so 库接口, 把 RKIV 烧录到 vendor storage:

```

int main(void)
{
    uint8_t rkiv[] = {0x00, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77,
                      0x88, 0x99, 0xaa, 0xbb, 0xcc, 0xdd, 0xee, 0xff};

    hdcp_rkiv_write(rkiv, 16);

    return 0;
}

```

3. 如果烧录 HDCP2x RX KEY，需要重新再次烧录 RKIV.

4. hdcp2\_tx\_rx 需要更新到 V3.4 及以上版本。

```

rk3588:/ # logcat | grep HDCP2
03-16 07:36:03.988 347 347 D HDCP2 : hdcp2 version v3.4
03-16 07:36:03.988 347 347 D HDCP2 : HDCP Port: 1
03-16 07:36:03.991 347 347 D HDCP2 : Firmware: [1]

```

## HDCP1.4 配置及说明

### HDMIRX

#### dts 配置

- 单独支持 HDCP1.4:

```

&hdmirx_ctrler {
    status = "okay";
    hdcp1x-enable;
};

```

- 如果使能 HDCP2.3，也会默认使能HDCP1.4:

```

&hdmirx_ctrler {
    status = "okay";
    hdcp2x-enable;
};

```

### Key 烧写

#### RK 工具烧写

Key 拆分和转换工具从SDK获取：

RKTools/windows/Rockchip\_HdcpKey\_Writer\_v1.0.1.7z 工具包下面有 KeyConvertor key 的拆分和转换工具，更新到 KeyConvertor-v2.0.5 可以拆分成一个bin文件只有一个key.

Key 烧写工具从SDK获取：

RKTools/windows/Rockchip\_HdcpKey\_Writer\_v1.0.1.7z

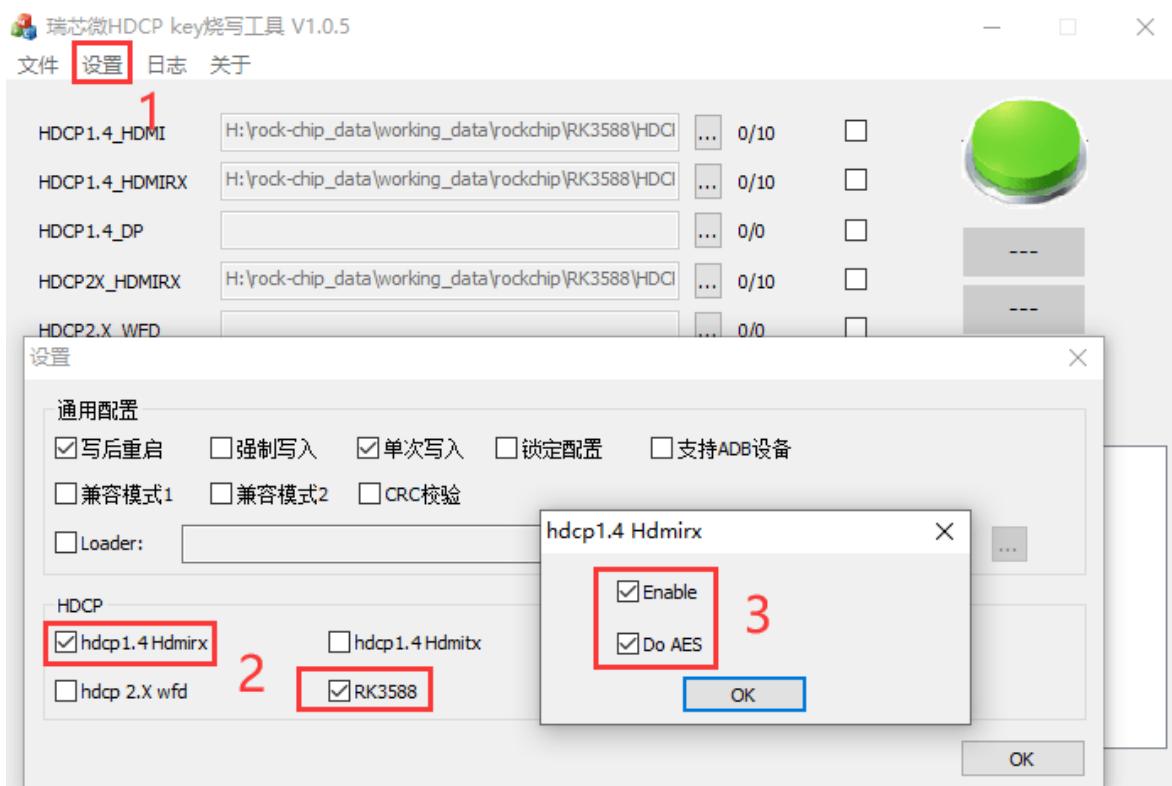
- 先用 Key 拆分和转换工具，从原始的 Key 文件中拆出部分 Key，然后把该 Key 转成 .skf 后缀的烧写Key.

- 如果不用RK提供的工具烧写，需要把Key拆成一个文件只包含一个Key，可以在提取原始Key的“每文件KEY个数”填入1，点提取即可生成单独的Key文件，不需要转成.skf格式。



- 机器进入Loader模式，烧写工具勾选 "hdcp1.4Hdmirx" 和 "RK3588" 选项，RK3588 HDCP1.4 Key 增加了 AES 加密，所以 "Do AES" 为可选项，不选的话只有 SEED 加密，选上的话会在 SEED 加密的基础上再进行 AES 加密，更安全。然后导入上面的.skf 文件，点击写入即可完成 Key 的烧写。

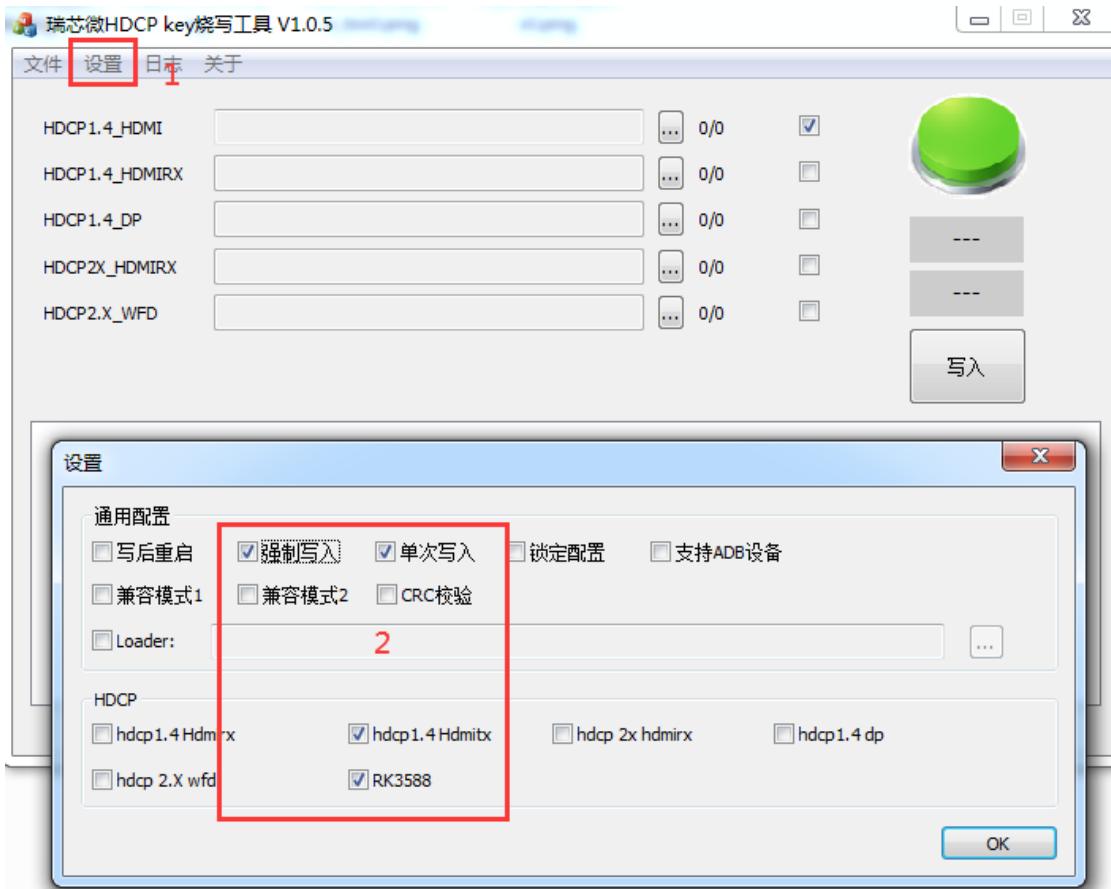
说明：工具写完一个 Key 之后自动会跳到下一个 Key。



可以自己写应用，调用 libhdcp.so 接口，对 KEY 进行加密并烧写对 Vendor Storage.

## HDMITX

- HDMITX HDCP 1.4 不需要特别配置 dts，驱动默认注册。
- 要正常使用首先需要分割、转换 KEY，所用的工具与方法与 HDMIRX 一致，参考 3.1.2.1 章节。



- 烧写 KEY 所用工具也与 RX 一致。机器进入 Loader 模式，烧写工具点击设置，勾选 "hdcp1.4 Hdmitx"、"RK3588"、"强制写入"、"单次写入" 选项。然后导入 key 转换后的 .skf 文件，点击写入即可完成 Key 的烧写。
- 说明：工具写完一个 Key 之后自动会跳到下一个 Key。

## DPTX

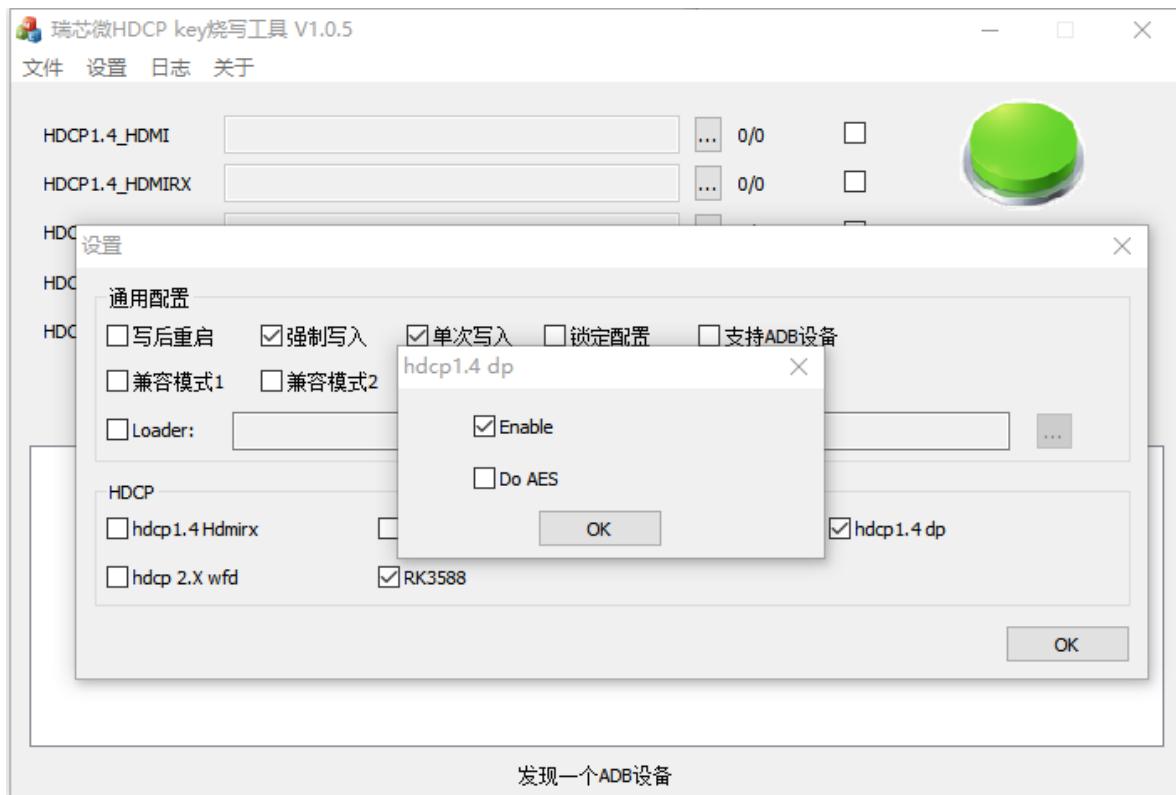
### dts 配置

默认 DPTX 的驱动默认支持 HDCP1.4, 不需要额外的 dts 配置。

### Key 烧写

Key 拆分和转换工具，与 HDMI RX 一致，参考 3.1.2.1 章节。

烧写 KEY 所用工具也与 HDMI RX 一致。机器进入 Loader 模式，烧写工具点击设置，勾选 "hdcp1.4 dp"、"RK3588"、"强制写入"、"单次写入" 选项。然后导入 key 转换后的 .skf 文件，点击写入即可完成 Key 的烧写。另外同 RK3588 HDCP1.4 Key，增加了 AES 加密，所以 "Do AES" 为可选项，不选的话只有 SEED 加密，选上的话会在 SEED 加密的基础上再进行 AES 加密。



## HDCP2.3 配置及说明

### HDMIRX

#### dts 配置

- 使能 HDCP2.3，也会默认使能HDCP1.4:

```
&hdcp1 {  
    status = "okay";  
};  
  
&hdmirx_ctrller {  
    status = "okay";  
    hdcp2x-enable;  
};
```

### 打包 Firmware 和 启动服务

- DWC\_HDCP2\_BASE\_ESM\_Firmware 解压到 Linux 环境下，然后把 hdcp\_receivers.bin 拷贝到根目录和 tools/ 目录下，如果同时支持HDMITX的话，可以同时把 hdcp\_transmitter.bin 也拷贝到相同的目录下。

```
DWC_HDCP2_BASE_ESM_Firmware$  
cp ./hdcp_receivers.bin .  
cp ./hdcp_receivers.bin tools/
```

- 打包 Firmware 和 RX KEY，执行 **build\_rockchip\_fw.sh** 脚本

```
DWC_HDCP2_BASE_ESM_Firmware$ ./build_rockchip_fw.sh
```

打包过程会有几个步骤会提示输入的：

1. 选择打包固件类型，选择: 1 -build firmware for both HDMIRX and HDMI TX, 生成  
`./firmware/hdpc2_hdmi.fw` 固件。

```
1 -build firmware for both HDMIRX and HDMI TX
2 -build firmware for HDMI TX only
3 -build firmware for DP
Choose the fw type: 1
```

2. 需要创建 RX KEY 的数量，根据需求创建，比如10或是100等，生成单独的 KEY 存放到  
`./rxkeys/hdcpkeys?-?/` 目录下的 `fw_hdcp_receivers_*.bin` 文件，如果客户用自己的工具烧写  
 的话，可以直接烧写 `fw_hdcp_receivers_*.bin` 文件。

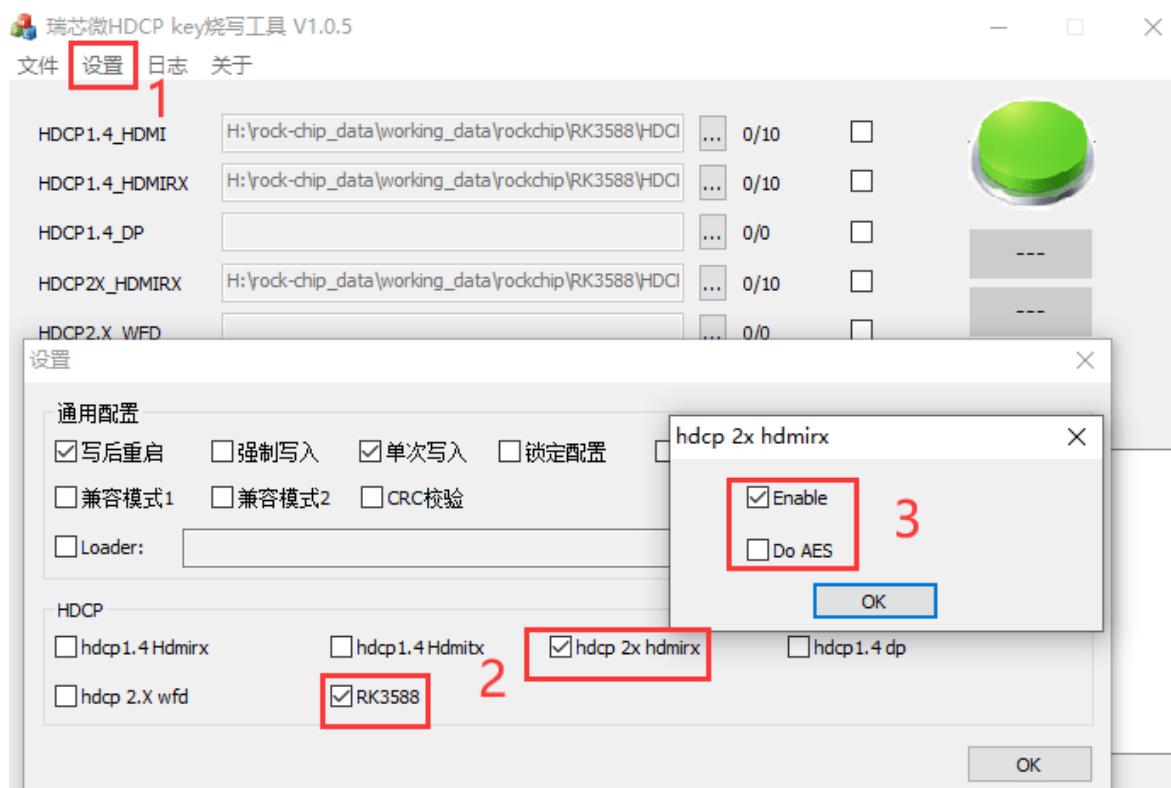
```
Create Rx Key Number:
10
.....
Create 10 keys to tools/rxkeys/hdcpkeys1-10/
```

3. 如果需要用 RK 提供的工具烧写 KEY，需要打包成 .skf 文件，存放到 `./rxkeys/hdcpkeys?-?/` 目录  
 下，如果不需要 RK 的工具烧写，这边可以选择 n.

```
Do want to pack these keys to .skf file? [y/n]:
y
10 file packed to ./rxkeys/hdcpkeys1-10/hdcprxkeys1-10.skf
```

- HDCP2.3 RX KEY 的烧写

1. 把 `hdcp2_hdmi.fw` 放到 `device/rockchip/rk3588/` 工程目录下，编译的时候会拷贝到 `vendor/firmware/hdcp2_hdmi.fw`。
2. 用 `RKDevInfoTool.exe` 工具，选择 `HDCP2X_HDMIRX` 导入上面生成的 `hdcprxkeys1-10.skf`，进  
 入 `Loader` 模式进行烧写。写完一个 Key，工具会自动跳到下一个 Key 等待烧写。
3. HDCP2.3 Key 不需要AES加密，所以这边只需要勾选 "RK3588" 和 "hdcp2x hdmirx"。



- hdcp2\_rx\_tx 服务

开机自动加载服务，如果出现认证异常，可以logcat | grep HDCP2 查看对应的log。

## HDMITX

### dts 配置

- HDMITX 与 HDMIRX 共用一个 HDCP 模块，使能 HDCP2.3，也会默认使能HDCP1.4:

```
&hdcp1 {
    status = "okay";
};
```

HDMI 的使能请参考《Rockchip\_Developer\_Guide\_HDMI\_CN》。

### 打包 Firmware 和 启动服务

- DWC\_HDCP2\_BASE\_ESM\_Firmware 解压到 Linux 环境下，然后把 hdcp\_transmitter.bin 拷贝到根目录和 tools/ 目录下，如果同时支持 HDMIRX 的话，可以同时把 hdcp\_receivers.bin 也拷贝到相同的目录下。

```
DWC_HDCP2_BASE_ESM_Firmware$  
cp ./hdcp_receivers.bin .  
cp ./hdcp_receivers.bin tools/
```

- 打包 Firmware 和 TX KEY，执行 **build\_rockchip\_fw.sh** 脚本

```
DWC_HDCP2_BASE_ESM_Firmware$ ./build_rockchip_fw.sh
```

打包过程会有几个步骤会提示输入的：

选择打包固件类型，如果同时支持 HDMI TX/RX 选择: 1 -build firmware for both HDMIRX and HDMITX，如果只支持 HDMITX，选择: 2 -build firmware for HDMITX only。最终生成 **./firmware/hdpc2\_hdmi.fw** 固件。

```
1 -build firmware for both HDMIRX and HDMITX  
2 -build firmware for HDMITX only  
3 -build firmware for DP  
Choose the fw type: 2
```

- hdcp2\_rx\_tx 服务

开机自动加载服务，如果出现认证异常，可以logcat | grep HDCP2 查看对应的log。

## DPTX

### dts 配置

DPTX 使用 HDCP0 模块，来使用 HDCP2.3 功能：

```
&hdcp0 {
    status = "okay";
};
```

- 使能 HDCP2.3，也会默认使能HDCP1.4:

## 打包 Firmware 和 启动服务

- DWC\_HDCP2\_BASE\_ESM\_Firmware 解压到 Linux 环境下，然后把 hdcp\_transmitter.bin 拷贝到根目录和 tools/ 目录下。

```
DWC_HDCP2_BASE_ESM_Firmware$  
cp ./hdcp_receivers.bin .  
cp ./hdcp_receivers.bin tools/
```

- 打包 Firmware 和 TX KEY，执行 **build\_rockchip\_fw.sh** 脚本

```
DWC_HDCP2_BASE_ESM_Firmware$ ./build_rockchip_fw.sh
```

打包过程会提示输入：

选择打包固件类型，选择: 3 -build firmware for DP，生成 **./firmware/hdpc2\_dp.fw** 固件。

```
1 -build firmware for both HDMIRX and HDMIITX  
2 -build firmware for HDMIITX only  
3 -build firmware for DP  
Choose the fw type: 3
```

对于 android 系统，要把 hdcp2\_dp.fw 拷贝到 /vendor/firmware/hdcp2\_dp.fw，可以在 hardware/rockchip/hdcp2 下加入如下补丁：

```
diff --git a/device.mk b/device.mk  
index 6d6c328..d7ba0a2 100644  
--- a/device.mk  
+++ b/device.mk  
@@ -82,6 +82,12 @@ ifeq ($(strip $(USE_PRODUCT_HDCP2_HDMI_FW)), true)  
    $(LOCAL_PATH)/hdcp2_hdmi.fw:vendor/firmware/hdcp2_hdmi.fw  
endif  
  
+USE_PRODUCT_HDCP2_DP_FW := $(shell test -f $(LOCAL_PATH)/hdcp2_dp.fw && echo  
true)  
+ifeq ($(strip $(USE_PRODUCT_HDCP2_DP_FW)), true)  
+    PRODUCT_COPY_FILES += \  
+        $(LOCAL_PATH)/hdcp2_dp.fw:vendor/firmware/hdcp2_dp.fw  
+endif  
+  
#  
# add Rockchip properties here  
#
```

并把 hdcp2\_dp.fw 放到 device/rockchip/rk3588/ 工程目录下，编译的时候会拷贝到 /vendor/firmware/hdcp2\_dp.fw。

对于 linux 系统，要把 hdcp2\_dp.fw 拷贝到 /system/etc/firmware/hdcp2\_dp.fw。

- hdcp2\_rx\_tx 服务

对于 android 系统，在 hardware/rockchip/hdcp2 下加入如下补丁：

```
diff --git a/init.hdcp2-host.rc b/init.hdcp2-host.rc
index 5a9bee8..57eef7e 100644
--- a/init.hdcp2-host.rc
+++ b/init.hdcp2-host.rc
@@ -1,7 +1,14 @@
 #on early-boot
 on post-fs
+    start hdcp2_dp
    start hdcp2_hdmi

+service hdcp2_dp /vendor/bin/hdcp2_tx_rx 0
+    class core
+    oneshot
+    disabled
+    group root system net_admin net_bt
+    capabilities DAC_OVERRIDE

service hdcp2_hdmi /vendor/bin/hdcp2_tx_rx 1
    class core
```

即可在系统开机时自动加载服务，如果出现认证异常，可以logcat | grep HDCP2 查看对应的log.

## HDCP 状态查看

### HDMIRX

```
cat /sys/class/misc/hdmirx_hdcp/status
```

```
HDCP Disable: HDCP 没使能
HDCP1.4: Authenticated start: 认证过程中
HDCP1.4: Authenticated success: 认证成功
HDCP1.4: Authenticated failed: 认证失败
HDCP1.4: Unknown status: 未知状态
```

```
HDCP Disable: HDCP 没使能
HDCP2.3: Authenticated success: 认证成功
HDCP2.3: Authenticated failed: 认证失败
HDCP2.3: No dectypted: 源端没有开启HDCP2.x
```

Source 没有开启 HDCP 加密，或是没有接入信号源：

```
HDCP2.3: No dectypted
HDCP1.4: Unknown status
```

### HDMITX

HDMITX 状态需要通过 DRM PROPERTY 进行查看，以 modetest -c 命令查看属性举例：

```
-----
Content Protection:
    flags: enum
        enums: Undesired=0 Desired=1 Enabled=2
```

```
value: 2
```

Undesired: 关闭hdcp。

Desired: 开启hdcp。

Enabled: hdcp已经开启并认证成功。

---

HDCP Content Type:

```
flags: enum
```

```
enums: HDCP Type0=0 HDCP Type1=1
```

```
value: 0
```

HDCP Type0:HDCP1.4 或 HDCP2.3 任一等级认证成功，都允许输出正常画面。

HDCP Type1: 必须HDCP2.3 认证成功，才能够输出正常画面。

注：该功能目前版本驱动未实际支持，默认任何等级认证成功都能输出正常，所以无需配置。

---

hdcp\_encrypted:

```
flags: range
```

```
values: 0 2
```

```
value: 2
```

hdcp认证等级：

0: hdcp未认证。

1: hdcp1.4。

2: hdcp2.3。

## DPTX

查看方式同 HDMI TX

## 常见问题分析

### hdcp2\_tx\_rx 启动失败

查看 log:

```
logcat | grep HDCP
```

```
Failed to open firmware file [vendor/firmware/hdcp2_hdmi.fw] //说明firmware没有导入
```

```
Failed to open firmware driver /dev/hl_dev1 //说明 hdcp 驱动没有加载, dts 需要使能对  
应的 hdcp1节点
```

```
07-14 10:53:17.033 2042 2042 D HDCP2      : hdcp2 version v3.0
07-14 10:53:17.033 2042 2042 D HDCP2      : HDCP Port: 1
07-14 10:53:17.034 2042 2042 D HDCP2      : Firmware: [1]
07-14 10:53:17.034 2042 2042 D HDCP2      :                      223664 bytes
07-14 10:53:17.034 2042 2042 D HDCP2      : check rx key ok
07-14 10:53:17.034 2042 2042 D HDCP2      : rxkey read from vendor success,
rxkey_len = 1000
07-14 10:53:17.034 2042 2042 D HDCP2      : Host Library version v0201
07-14 10:53:17.035 2042 2042 D HDCP2      : Exiting with err=1
```

如果出现上面这种没有报错，直接退出的情况，可以手动执行，看下对应的报错：

```
console:/ # ./vendor/bin/hdcp2_tx_rx 1 &
[1] 2799
console:/ # device name = /dev/hl_dev1
<INFO> [external/hdcp/host_library/core_lib/host_lib_core.c:1199] [PID:2799]
hlInst->driver->instance [0xee2401d0]
<INFO> [external/hdcp/host_library/core_lib/host_lib_core.c:1200] [PID:2799]
hlInst->driver [0xff8088f0:223664]
<WARN> [external/hdcp/host_library/core_lib/host_lib_core.c:1206] [PID:2799]
Error in file:external/hdcp/host_library/core_lib/host_lib_core.c, functio]

<ERROR> [external/hdcp/host_library/core_lib/host_lib_core.c:137] [PID:2799]
hostlib_init_mem failed
Failed HLC_Initialize, error -6
```

两面上面这种log，是固件打包问题，可能是 KEY 不对或是 DUK/PKF配置不对，又或是打包的方式不对，参考[HDCP 2.3 固件打包说明]

HLC\_HDCPRX\_LoadRxDwb failed //说明是RX KEY 不对

## HDMIRX 黑屏或是绿屏插拔无法恢复

- 先确认 dts 的 hdcp2x-enable 是否打开，可以 cat /sys/class/misc/hdmirx\_hdcp/enable 看值是否为 2。
- 如果 hdcp2x-enable 打开的话，那 hdcp2\_tx\_rx 这个服务必须要加载，不然就可能会出现黑屏或是绿屏，可以 logcat | grep HDCP2 进行确认。
- 如果 hdcp2\_tx\_rx 已经加载，那需要确认是否异常退出，同样 logcat | grep HDCP2 确认。
- echo 0 > /sys/class/misc/hdmirx\_hdcp/enable 之后恢复正常，可以确定是上面所说的问题。

## 同时打开 HDMI TX + HDMIRX 的 HDCP2.3, 拔出 RX 的时候 TX 会闪一下花屏

因为 HDMIRX 和 HDMI TX 共用 HDCP2.3 控制器，RX 在预览的时候拔出，会复位控制器，导致 HDCP2.3 控制器异常，hdcp2\_tx\_rx 检测到控制器异常之后，会重新初始化，重新做认证，所以会闪一下花屏，建议 TX + RX 的 HDCP2.3 不要同时打开。