密级状态：绝密（ ） 秘密（ ） 内部（ ） 公开（√）

Rockchip Android Widevine 开发指南

文件状态： [] 正在修改 [√] 正式发布	当前版本：	V1.3
	作　　者：	wd
	完成日期：	2017-09-22
	审　　核：	
	完成日期：	

福州瑞芯微电子有限公司

Fuzhou Rockchips Semiconductor Co . , Ltd

版本历史

版本号	作者	修改日期	修改说明	备注
V1.00	hjh	2017.04.14	基础版本	
V1.10	wd	2017.09.22	部分章节修改等	
V1.2	hjh	2017.11.08	1.文档结构调整，更符合开发流程。 2.通用性修改，使文档适用于不同 SDK	
V1.3	hjh	2017.11.08	修改 dts ion_drm 配置说明。	

目 录

版本历史 2
目 录 3
1. 概述 5
1.1. 本文档适用范围 5
1.2. 集成 Widevine 的软硬件要求 5
1.2.1. Widevine level 1 5
1.2.2. Widevine level 3 5
1.3. SVP（secure video path）音视频编码格式支持 5
1.4. WideVine 简介 5
1.5. WideVine 工作流程 6
2. Widwvine L1 开发 7
2.1. 项目开发流程 8
2.2. 方案框图 9
2.3. 需手动配置的部分 9
2.3.1. kernel 9
2.3.2. Android 10
2.3.3. u-boot 10
2.3.4. 需要关注的目录以及相关 commit 11
2.4. 编译 13
2.5. optee 环境测试 13
2.6. Device Provisioning keybox 烧写方案 15
2.6.1. keybox 申请流程 15
2.6.2. keybox 烧写方案框图 17
2.6.3. keybox 烧写步骤 17
2.6.4. Keybox 烧写工具以及文档获取 18

2.6.5. keybox 烧写出错排查 18
2.6.6. keybox API 介绍 18
2.7. 测试说明 22
2.7.1. 测试前准备 22
2.7.2. DRM Info 检测 23
2.7.3. exoplayer 码流播放 23
2.7.4. 厂测 23
2.7.5. log 提供 24
3. Widevine L3 开发 24
3.1. 测试前准备 24
3.2. DRM Info 检测 24
3.3. exoplayer 码流播放 24

1. 概述

1.1. 本文档适用范围

软件工程师、硬件工程师、测试工程师。

1.2. 集成 Widevine 的软硬件要求

1.2.1. Widevine level 1

- Android7.1 或者以上
- 内存 2G 或者以上
- 芯片支持 Trustzone
- 支持 secureboot（调试阶段可以不开启）
- 支持 HDCP

1.2.2. Widevine level 3

- 支持 secureboot（调试阶段可以不开启）

1.3. SVP（secure video path）音视频编码格式支持

VP9、H264、H265、VP8、MPEG2、MPEG4

1.4. WideVine 简介

Widevine 是 google 在 android 3.0 版本之后推出的一种 DRM 数字版权管理功能，官方网站为 http://www.widevine.com/，可以从该网站获取产品、技术、文档、keybox 及支持等信息。

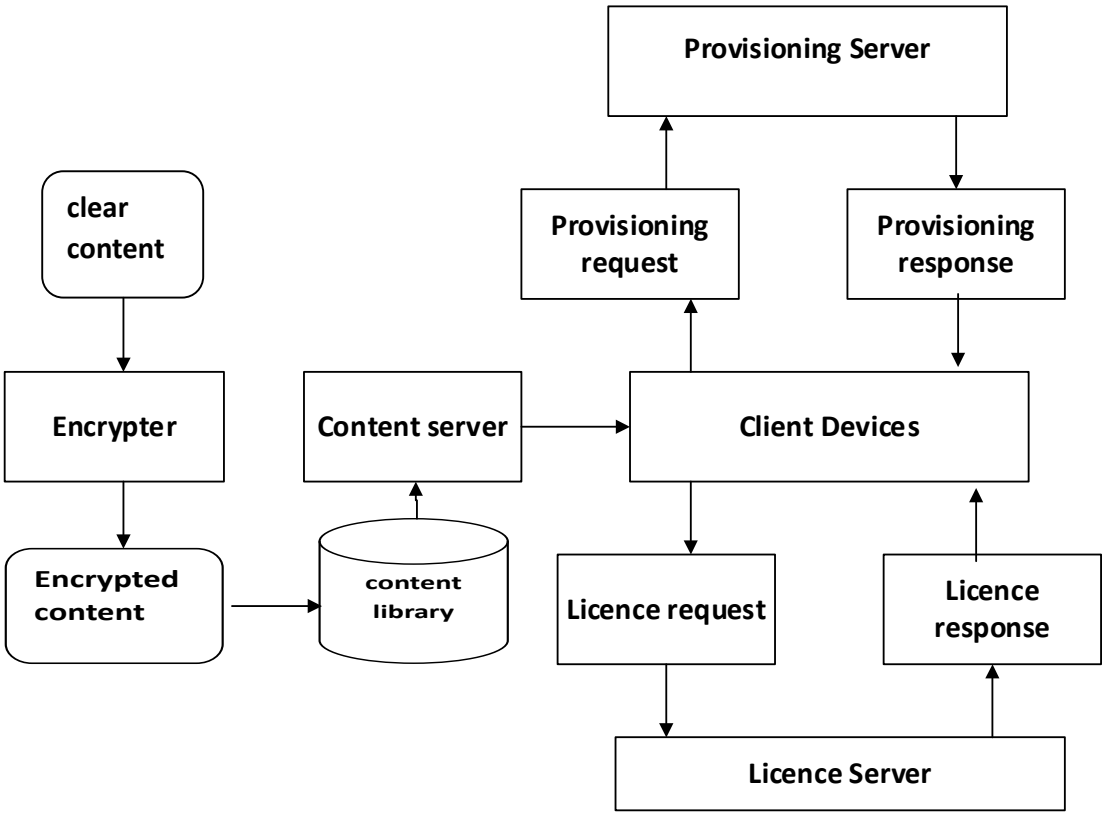
Widevine 有 3 种版本：

Security Level	Secure Boot Loader	Widevine Key Provisioning	Security Hardware or Trusted Execution Environment	Widevine Keybox and Video Key Processing	Hardware Video Path
Level 1	Yes	Factory	Yes	Keys never exposed in clear to host CPU	Hardware Protected Video Path
Level 2	Yes	Factory	Yes	Keys never exposed in clear to host CPU	Clear video streams delivered to renderer
Level 3	Yes	Field	No	Clear keys exposed to host CPU	Clear video streams delivered to decoder

由于 L2 无太多需求，目前主要提供 L1，L3 版本。

L3 版本能为内容和 keybox 提供基本和必要的保护，对芯片和方案没有特别要求，适用于内容提供商没有特殊要求的场景。L1 版本使用 Rockchip 芯片的 Trustzone 硬件保护机制，对 keybox、加解密密钥以及解密后的码流提供硬件级别的保护，L1 版本适用于内容提供商明确要求 Trustzone 特性和安全视频路径（Secure Video Path）的场景。

1.5. WideVine 工作流程



具体流程说明如下：

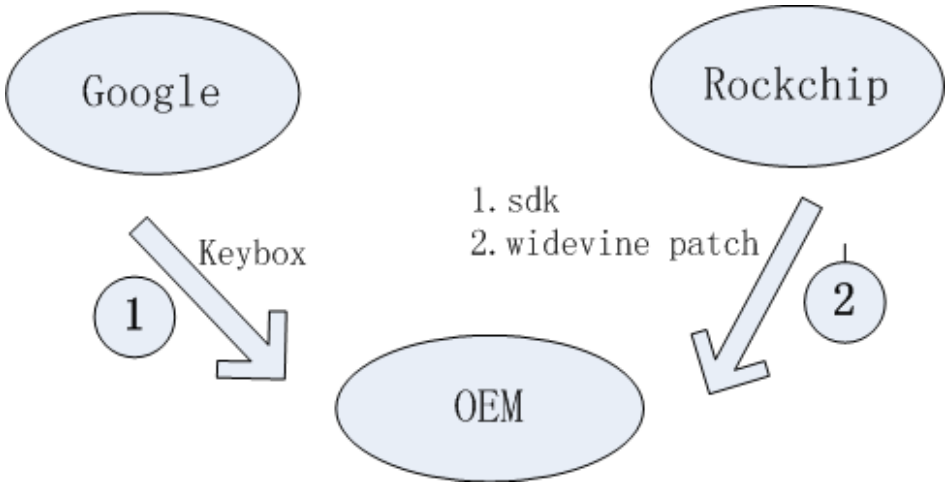
步骤 1 加密服务器把明文内容加密后存放于内容库，加密内容通过标准的 HTTP 服务器传送到客户端设备。

步骤 2 provisioning server 来分发设备唯一证书（这一步非必须，只是为了更加安全 OEMCrypto v12 or newer Android O 以后的版本必须 .），设备使用 Widevine keybox 作为信任根，从配置服务器获取证书，应用程序使用 RSA 签名的证书来获取许可证请求，而不是 Widevine Keybox。

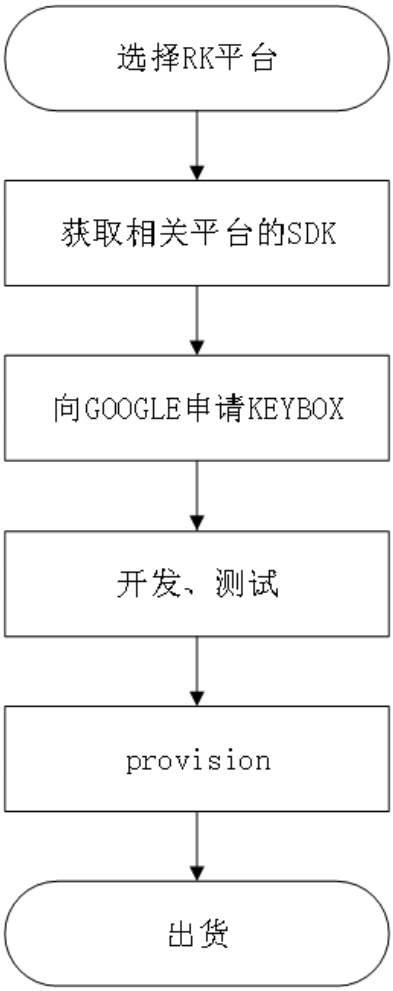
步骤 3 客户端向 licence server 请求，licence server 确认设备合法后，发送安全（签名&加密）licence 给客户端。只有当获取到 licence 后，设备才有权限访问相关 DRM 内容。

2. Widwvine L1 开发

项目关系图



2.1. 项目开发流程



步骤 1 选择 Rockchip 平台，若是 Widevine L1 版本，请选择支持 Trustzone 特性的平台。

步骤 2 从 Rockchip 获取支持 Widevine L1 的 Android SDK。

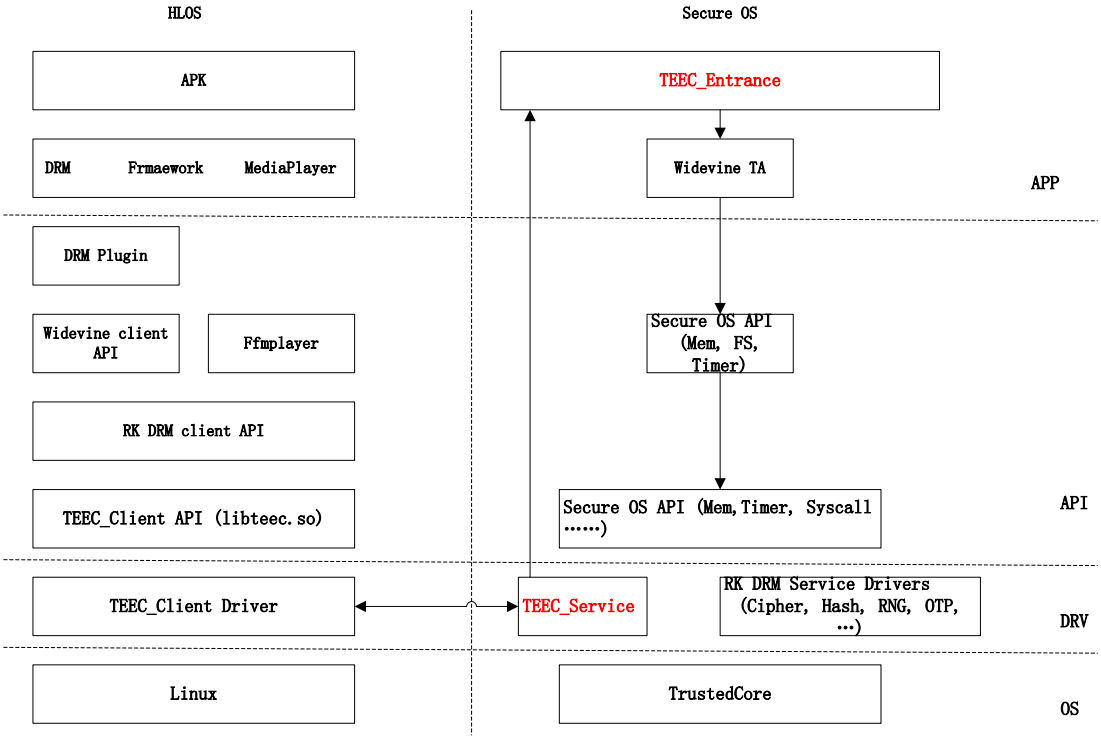
步骤 3 从 Google 获取 keybox。（参考 2.7.1 keybox 申请流程）

步骤 4 集成补丁，开发、测试。

步骤 5 工厂生产，完成 keybox 烧写。（参考 2.7.Device Provisioning keybox 烧写方案）

----结束

2.2. 方案框图



2.3. 需手动配置的部分

2.3.1. kernel

在项目对应的 dts 增加以下配置：

kernel 4.4 版本：

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

Kernel 3.10 版本

```
&ion_drm {
    reg = <0x20000000 0x10000000>; /* 256MB */
```

};

2.3.2. Android

2.3.2.1. /device/rockchip/common/BoardConfig.mk

// BUILD_WITH_WIDEVINE 这个如果在 vendor/widevine 目录有用到，就设置为 true,没有用到就不用设置。目前最新的 SDK 已经把这个去掉，就只用 BOARD_WIDEVINE_OEMCRYPTO_LEVEL 判断

```
#widevine configuration
BUILD_WITH_WIDEVINE ?= true
# for widevine drm
BOARD_WIDEVINE_OEMCRYPTO_LEVEL := 1
```

2.3.2.2. /device/rockchip/rkxxxx(rkxxxx 指的是项目目录，比如：rk3399，rk3328 或者客户自定义的项目目录)

```
#for optee support
PRODUCT_HAVE_OPTEE ?= true
```

2.3.2.1 以及 2.3.2.2 编译变量最终是否配置成功，请通过以下 get_build_var 命令确认，如

```
AndroidN/rk3328$ get_build_var PRODUCT_HAVE_OPTEE
true
```

2.3.3. u-boot

Rk3399：

```
commit 45a462c385765078ec57c8b4de13025dc403e792
Author: Zhang Zhijie <zhangzj@rock-chips.com>
Date:   Fri Oct 20 09:59:19 2017 +0800

    rk3399: bl32: update version to 1.11
```

如果 u-boot 包含以上提交或者更新的版本，则无需做任何修改。

如果 u-boot 不包含以上提交，则需要

先确认 u-boot 的 git log 中含有以下 commitID

```
commit 81ceebdf5a73e2e14facb7ad082032149f81fecd
```

然后再修改/u-boot/include/configs/rk33plat.h 文件

#define CONFIG_MERGER_TRUSTIMAGE_DRM

其他 SDK：无需特殊配置

2.3.4. 需要关注的目录以及相关 commit

如果 2.3.1-2.3.2 这些修改都确认修改完毕，Widevine L1 的相关配置就已经完成，可以进行相关的 Widevine L1 的测试。如果测试中出现问题，可以先对下面的目录中的提交进行确认，需要确保这些提交已经在 SDK 中。

2.3.4.1. /kernel

Kernel 4.4 版本：

commit 1b974395a86c31503ddb03c70e5d7a2787550b41
Author: rimon.xu <rimon.xu@rock-chips.com>
Date: Thu Aug 31 18:51:07 2017 +0800
video: rockchip: vpu: Add support video secure access

commit 9cd28142506c54e5de1f6f3387f4537ca8c6faa2
Author: rimon.xu <rimon.xu@rock-chips.com>
Date: Fri Sep 1 10:32:50 2017 +0800

arm64: dts: rockchip: rk3399-android: add secure memory for drm.

Kernel 3.10 版本：

Author: Jianqun Xu <jay.xu@rock-chips.com>
Date: Tue Nov 7 14:04:49 2017 +0800
rk: ion: check valid for size of drm heap create

commit 05924704ff5cc93d691a65687f9fbc817e388622
Author: Jianqun Xu <jay.xu@rock-chips.com>
Date: Tue Nov 7 11:15:05 2017 +0800
rk: ion: rockchip: continue to create if someone heap failed
If one heap fail to create, the rockchip ion driver will fail.
Actually we should allow other heaps who created successfully.

The ion_heap_destroy hasn't check if the heap is valid or not,
so let's do it before that callback.

commit 13ea80364c13aced08e12af2ffdfd514452d709a
Author: Xuhanrui <xhr@rock-chips.com>
Date: Fri Oct 27 14:33:18 2017 +0800

arm64: dts: rk322xh: add ion drm heap support
Default to set 0M for drm heap, someone should re-defined
the reg in dts file, such as
&ion_drm {
reg = <0x20000000 0x10000000>; /* 256MB */

2.3.4.2. /hardware/rockchip/librkvpu

commit da8f8032916ffaa281d70b85bc82444b50dd9247
Author: rimon.xu <rimon.xu@rock-chips.com>
Date: Wed Aug 23 10:04:36 2017 +0800

[drm]: add VPU_API_CMD for drm svp.

2.3.4.3. /hardware/rockchip/omx_il

commit c96bc9a44225ce1bb60e00b5305e5b2a30611068
Author: rimon.xu <rimon.xu@rock-chips.com>
Date: Tue Aug 22 10:51:25 2017 +0800

[drm]: when svp access, force to mpp

commit 52271b828fa7f56483f5464c227bb6334cee7616
Author: rimon.xu <rimon.xu@rock-chips.com>
Date: Thu Aug 17 19:38:19 2017 +0800

[SVP]: add svp for first version

2.3.4.4. /vendor/rockchip/common/vpu

commit 36c6b6d8706eddfd90a37fc63d272152902b2164
Author: Rimon Xu <rimon.xu@rock-chips.com>
Date: Wed Sep 13 09:48:52 2017 +0800

[drm]: update mpp for h264 svp

commit f8b4f28dd6a2a77780dbc3d2653ead8ef30b8c08
Author: rimon.xu <rimon.xu@rock-chips.com>
Date: Tue Aug 22 09:29:33 2017 +0800

[drm]: add secure decode component for rk3399 7.0

2.3.4.5. /vendor/widevine

commit 5a67a8b401086f440ef765ae51aae96fd6460d44
Author: willam.wei <willam.wei@rock-chips.com>
Date: Fri Sep 1 09:21:28 2017 +0800

widevine level 1 support

2.4. 编译

参考 SDK 发布文档进行编译。

2.5. optee 环境测试

widevine L1 要基于 TEE 环境实现，测试前请先确认 TEE 环境运行正常。
测试方法如下
将 SDK/vendor/rockchip/common/security/optee/optee_test/
目录中的文件 push 到开发板的对应目录：

- testapp、testapp_storage 这两个文件 push 到开发板 /system/bin/目录。
- 8cccf200-2450-11e4-abe20002a5d5c52c.ta （对应 testapp）、
  8dddf200-2450-11e4-abe20002a5d5c53d.ta（对应 testapp_storage）
  这两个文件 push 到开发板 /system/lib/optee_armtz/目录（如果开发板没有这个目录的话需要创建一个）。

然后可以执行 testapp、testapp_storage 进行测试。

ADB 结果参考：

```
rk3399_box:/ # testapp
test value : Pass!
test tembuf : Pass!
test shmbuf : Pass!

rk3399_box:/ # testapp_storage
Success! Please check details from the serial!
```

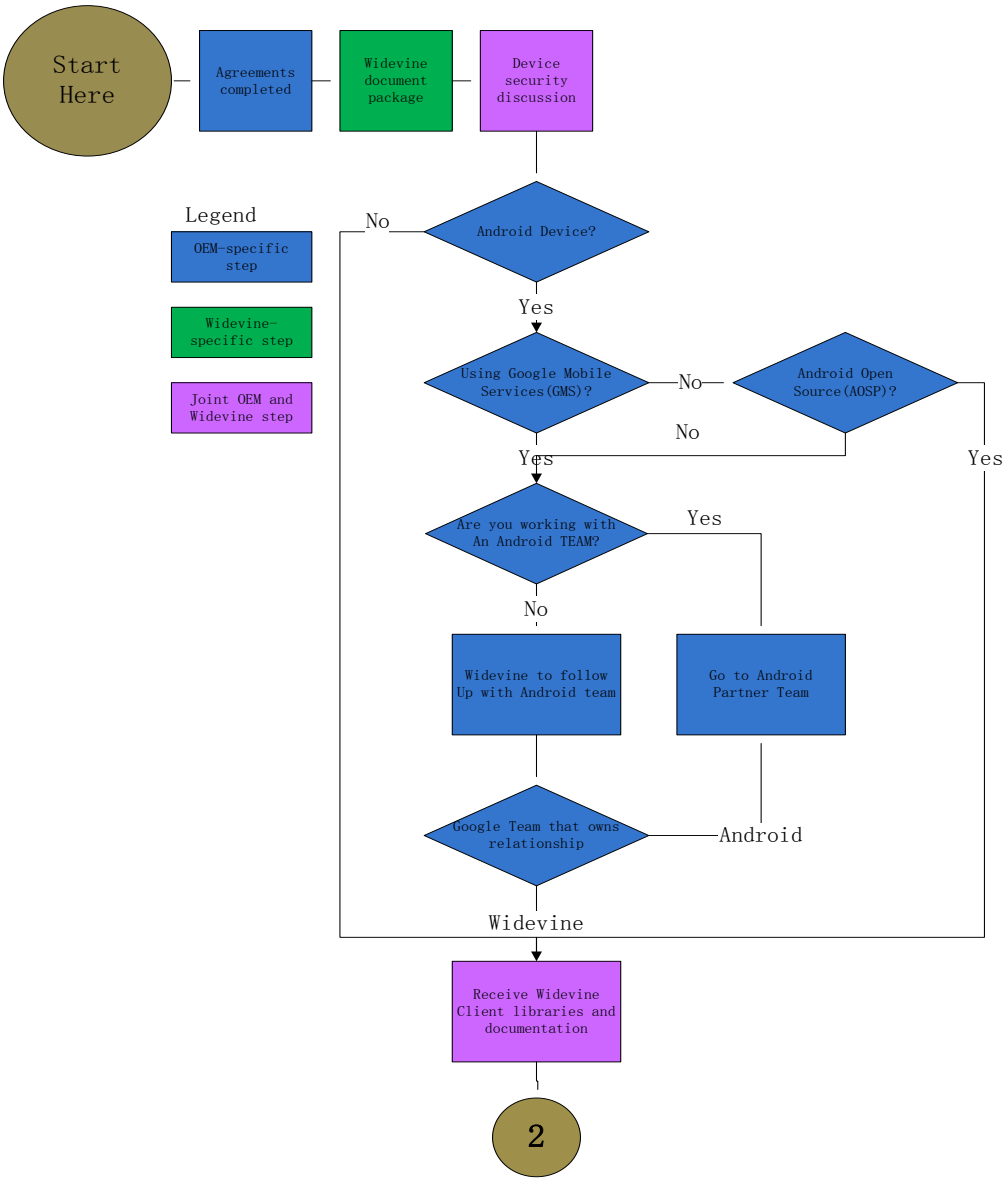
串口结果参考：

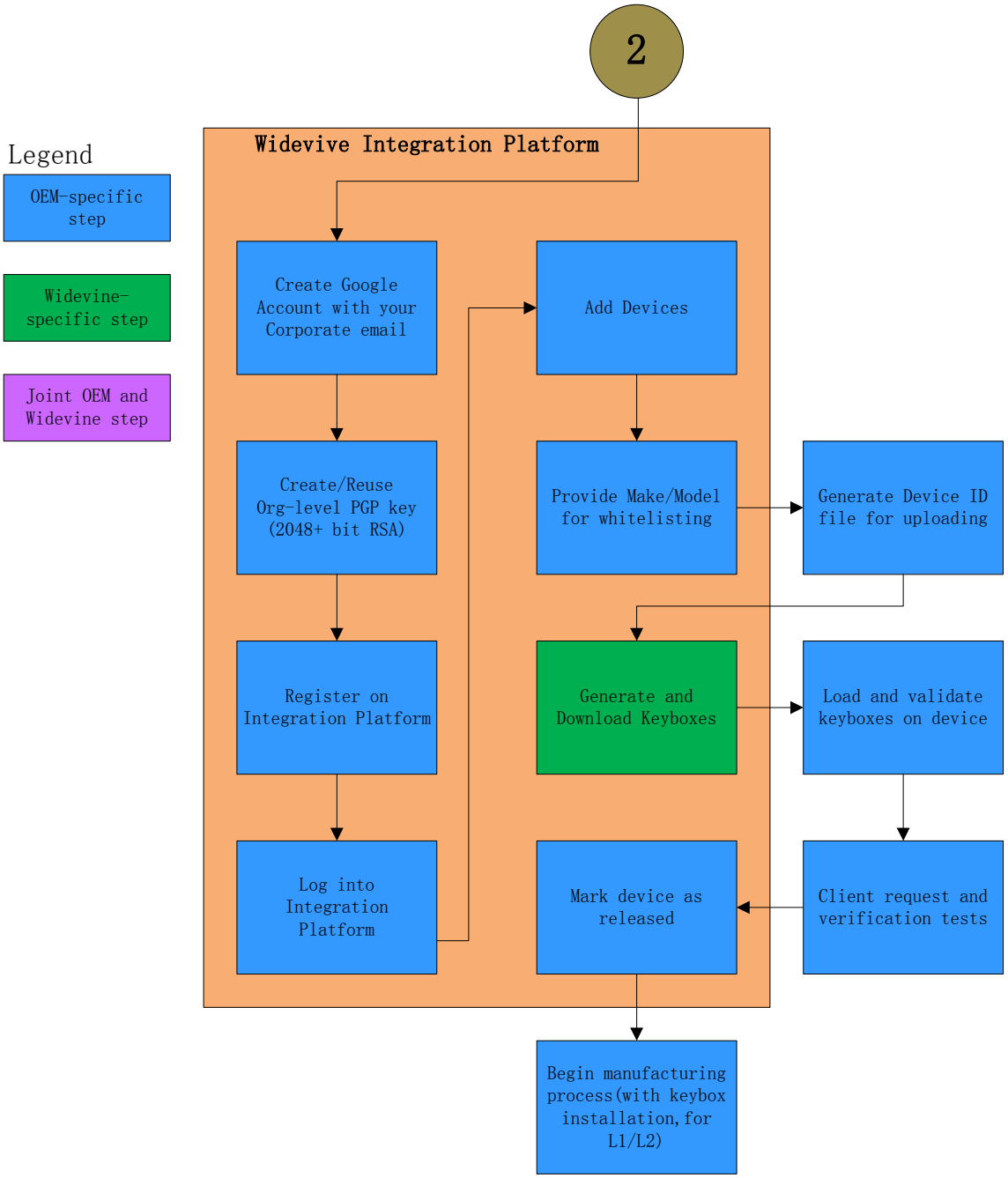
```
INF USER-TA:TA_OpenSessionEntryPoint:61: Hello Test App!
INF USER-TA:TA_InvokeCommandEntryPoint:110: membuf test : Pass!
INF USER-TA:TA_CloseSessionEntryPoint:73: Goodbye Test App!

INF USER-TA:TA_OpenSessionEntryPoint:58: Hello Test Storage!
INF USER-TA:TA_InvokeCommandEntryPoint:120: TEE_CreatePersistentObject success !
INF USER-TA:TA_InvokeCommandEntryPoint:127: TEE_WriteObjectData success !
INF USER-TA:TA_InvokeCommandEntryPoint:134: TEE_SeekObjectData success !
INF USER-TA:TA_InvokeCommandEntryPoint:141: TEE_ReadObjectData success !
INF USER-TA:TA_InvokeCommandEntryPoint:144: TA Storage : verify success
INF USER-TA:TA_InvokeCommandEntryPoint:148: before TEE_CloseAndDeletePersistentObject
INF USER-TA:TA_InvokeCommandEntryPoint:150: after TEE_CloseAndDeletePersistentObject
INF USER-TA:TA_CloseSessionEntryPoint:70: Goodbye Test Storage!
```

2.6. Device Provisioning keybox 烧写方案

2.6.1. keybox 申请流程





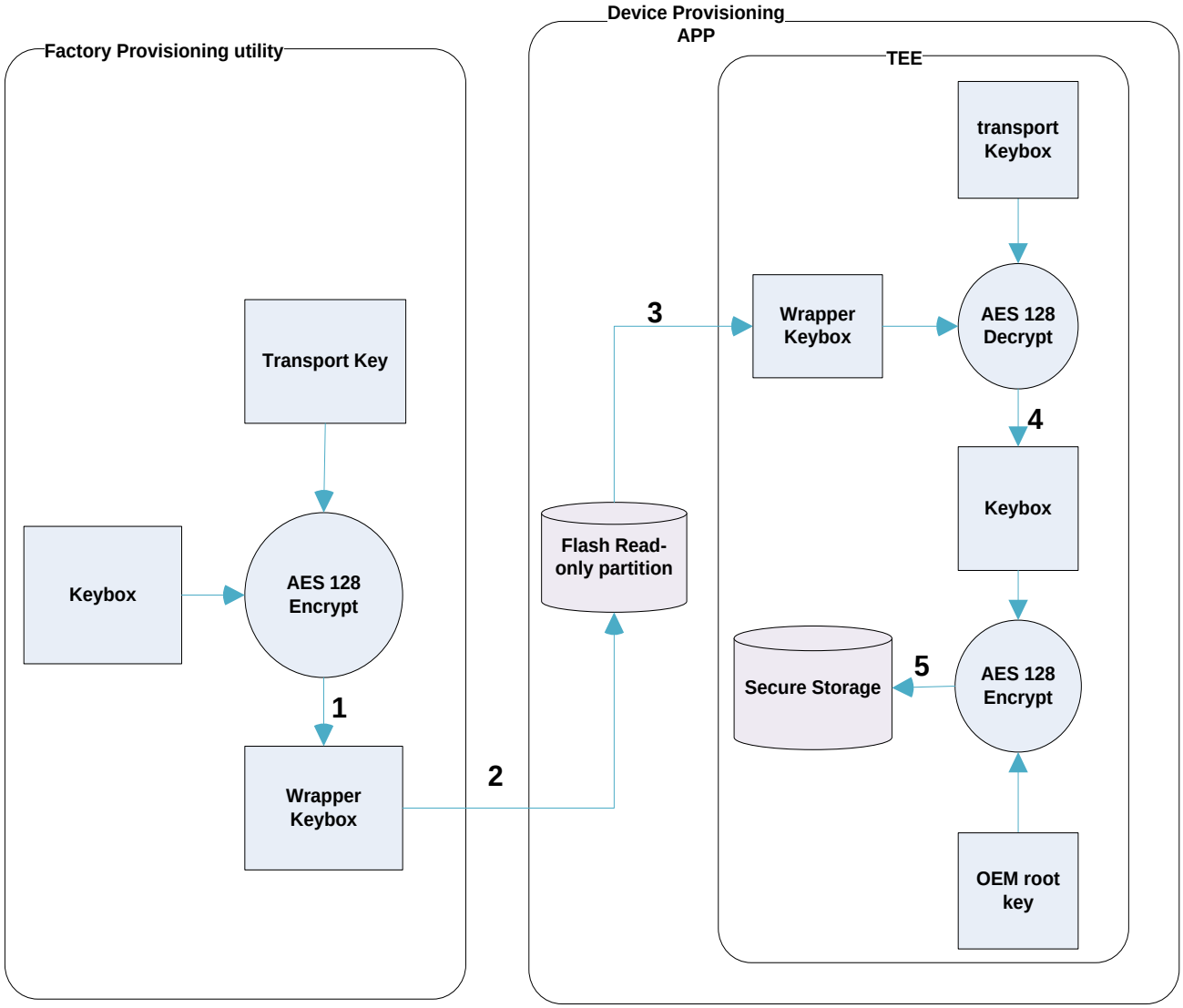
详细申请说明请参考文档

Widevine_DRM_Getting_Started_Device_Management.pdf

该文档以及 keybox 申请请联系 widevine

http://www.widevine.com/contact.html

2.6.2. keybox 烧写方案框图



2.6.3. keybox 烧写步骤

步骤 1 OEM 厂商使用 RK 提供的 PC 端工具 rkpacker 加密 keybox。

步骤 2 OEM 厂商使用 RK 提供的 PC 端工具 KeyboxWriter 把机密后的 keybox 以及密钥写入到 vendor 区，KeyboxWriter 请以管理员权限打开。

步骤 3 写入 vendor 区成功后，系统自动重启，然后通过 TEE 把 keybox 搬到 Secure storage，EMMC secure storage 目前用的是 RPMB。

2.6.4. Keybox 烧写工具以及文档获取

Keybox 烧写工具：RKTools/windows/ keybox 烧写工具 keywriter-v1.5.zip

keybox 烧写文档：RKDocs/RKTools manuals/ Rockchip_Provision_Tool 说明手册_V1.1_20171102.pdf

2.6.5. keybox 烧写出错排查

1. 向供应商确认使用的 EMMC 是否支持 RPMB。

2. keybox 烧写工具是否有以管理员权限打开。

3.确认 tee-supplicant 、rk_store_keybox 开机运行.

4.以上都确认没问题了，如果还是报错，请提供串口打印的 log 以及 logcat

2.6.6. keybox API 介绍

Keybox API 可用于客户自行设计 keybox 烧写工具或者厂测时校验 keybox 是否已经烧写以及最终烧写的 keybox 是否正确。

OEMCrypto_Initialize

OEMCrypto_Terminate

OEMCrypto_GetDeviceID

OEMCrypto_IsKeyboxValid

OEMCrypto_InstallKeybox

OEMCrypto_Initialize

OEMCryptoResult OEMCrypto_Initialize(void);

Initializes the crypto hardware.

Parameters

None

Returns

OEMCrypto_SUCCESS success

OEMCrypto_ERROR_INIT_FAILED failed to initialize crypto hardware

Threading

No other function calls will be made while this function is running. This function will not be called again before OEMCrypto_Terminate().

Version

This method is supported by all API versions.

OEMCrypto_Terminate

OEMCryptoResult OEMCrypto_Terminate(void);

Closes the crypto operation and releases all related resources.

Parameters

None

Returns

OEMCrypto_SUCCESS success

OEMCrypto_ERROR_TERMINATE_FAILED failed to deinitialize

crypto hardware

Threading

No other OEMCrypto calls are made while this function is running. After this function is called,

no other OEMCrypto calls will be made until another call to OEMCrypto_Initialize() is made.

Version

This method is supported by all API versions.

OEMCrypto_InstallKeybox

OEMCryptoResult OEMCrypto_InstallKeybox(

uint8_t *keybox, uint32_t keyboxLength);

Decrypts a wrapped keybox and installs it in the security processor. The keybox is unwrapped

then encrypted with the OEM root key. This function is called from the Widevine DRM plugin at

initialization time if there is no valid keybox installed. It looks for a wrapped keybox in the file

/factory/wv.keys and if it is present, will read the file and call

OEMCrypto_InstallKeybox() with

the contents of the file.

Parameters

[in] keybox pointer

to encrypted Keybox data as input

[in] keyboxLength length

of the keybox data in bytes

Returns

OEMCrypto_SUCCESS success

OEMCrypto_ERROR_BAD_MAGIC

OEMCrypto_ERROR_BAD_CRC

OEMCrypto_ERROR_INSUFFICIENT_RESOURCES

Threading

This function is not called simultaneously with any other functions.

Version

This method is supported in all API versions.

OEMCrypto_IsKeyboxValid

OEMCryptoResult OEMCrypto_IsKeyboxValid();

Validates the Widevine Keybox loaded into the security processor device. This method verifies

two fields in the keybox:

- Verify the MAGIC field contains a valid signature (such as, ‘k’ ’ b’ ’ o’ ’ x’).
- Compute the CRC using CRC32POSIX1003.2

standard and compare the checksum

to the CRC stored in the Keybox.

The CRC is computed over the entire Keybox excluding the 4 bytes of the CRC (for example,

Keybox[0..123]). For a description of the fields stored in the keybox, see Keybox Definition .

Parameters

none

Returns

OEMCrypto_SUCCESS

OEMCrypto_ERROR_BAD_MAGIC

OEMCrypto_ERROR_BAD_CRC

Threading

This function may be called simultaneously with any session functions.

Version

This method is supported in all API versions.

OEMCrypto_GetDeviceID

OEMCryptoResult OEMCrypto_GetDeviceID(

uint8_t* deviceID,

uint32_t *idLength);

Retrieve DeviceID from the Keybox.

Parameters

[out] deviceId pointer

to the buffer that receives the Device ID

[in/out] idLength – on input, size of the caller’s device ID buffer. On output, the number of bytes

written into the buffer.

Returns

OEMCrypto_SUCCESS success

OEMCrypto_ERROR_SHORT_BUFFER if the buffer is too small to return device ID

OEMCrypto_ERROR_NO_DEVICEID failed to return Device Id

Threading

This function may be called simultaneously with any session functions.

Version

This method is supported in all API versions.

2.7. 测试说明

请参考测试说明，使用 exoplayer、drminfo APK 进行测试。

首先确认网络能访问 Google 测试 服务器，中国大陆需要连接 vpn 测试。

2.7.1. 测试前准备

- 请确认 trust OS 运行正常

  可以参考 2.5.optee 环境测试

- 确认 keybox 已经烧写，正确烧写 keybox 的机器开机会串口打印以下信息：

```
INF USER-TA:rk_is_keybox_exist:486: keybox exists in secure storage.
```

或者 logcat 打印以下信息：

```
shell@rk322x_box:/ # logcat |grep keybox
I/rk_store_keybox(  138): keybox exists in secure storage.
```

- 请确设备以下文件存在:

  - /system/vendor/lib/liboemcrypto.so
  - /system/lib/libRkWvClient.so
  - /system/vendor/lib/mediadrm/libwvdrmengine.so
  - /system/lib/optee_armtz/c11fe8ac-b997-48cf-a28de2a55e5240ef.ta
  - /system/bin/rk_store_keybox

2.7.2. DRM Info 检测

步骤 1 运行 DRM InfoAPK（drminfo.apk)。

步骤 2 确认 Google Widevine Modular DRM 中 SecurityLevel 是否为 L1

----结束

2.7.3. exoplayer 码流播放

步骤 1 运行 exoplayerAPK（exoplayer.apk)。

步骤 2 选择列表中 Widevine 开头标题下的片源播放。

----结束

2.7.4. 厂测

工厂如果有需要在厂测工具加下 keybox 校验的功能，可以参考补丁包里面的 keyboxtest demo。

2.7.5. log 提供

测试过程有任何问题，请提供串口输出的 kernel、trust os log 以及 logcat。

3. Widevine L3 开发

3.1. 测试前准备

- 请确认是否有连接 vpn 测试，大陆网络无法访问测试片源。
- 请确认机器存在以下文件（根据平台差异，库文件在对应的 lib 目录或 lib64 目录）

  - /system/vendor/lib/mediadrm/libwvdrmengine.so
- 如果以上都确认 ok 还是无法播放，请提供 kernel log 以及 logcat

3.2. DRM Info 检测

步骤 1 运行 DRM InfoAPK（drminfo.apk)。

步骤 2 确认 Google Widevine Modular DRM 中 SecurityLevel 是否为 L3

----结束

3.3. exoplayer 码流播放

步骤 1 运行 exoplayerAPK（exoplayer.apk)。

步骤 2 选择 Widevine DASH Policy Tests(GTS) 下的片源播放，L3 的只能选择 HDCP not specified 和 HDCP not required 这 2 个测试。