Android_Widevine_ 开发指南

版本号: 1.0
发布日期: 2022.04.28

版本历史

版本号	日期	制/修订人	内容描述
1.0	2022.04.28	AWA1757	建立初始版本

目　录

插　图

1 概述

本文档主要介绍全志 Widevine 的工作原理以及开发过程及注意事项。

1.1 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

1.2 Widevine 简介

Widevine 是美国的一家专门提供流媒体数字版权保护 (DRM) 技术的公司，该公司的 DRM 技术被广泛地应用于数字流媒体领域，例如在线视频、数字电视等等。2010 年 9 月，谷歌收购了此公司，意图拓展自己的数字流媒体电影服务以及获得其 DRM 保护技术，谷歌在 Android 上搭载了 Widevine 的 DRM 技术，已经成为了 GMS(Google Mobile Service) 中必备的内容。

1.2.1 Widevine 整体框架

Widevine 是 DRM 的一种插件，在 Android 中应用，使用的架构是固定的 DRM 架构：

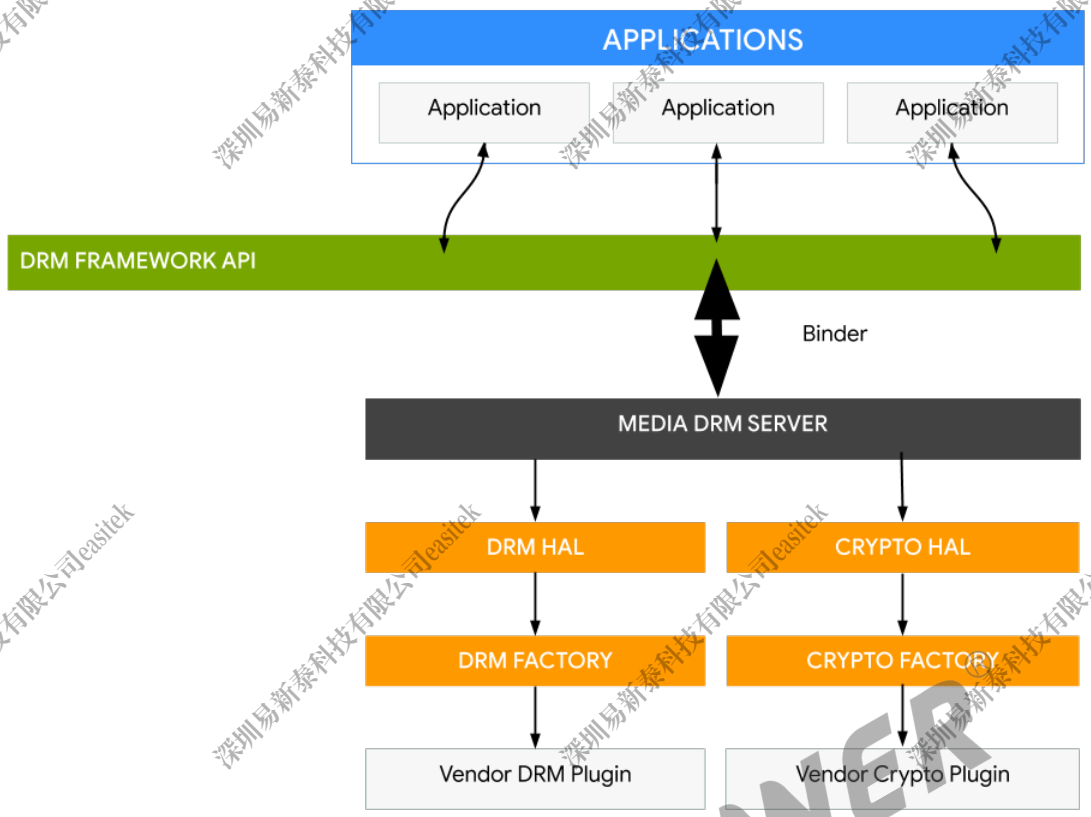

图 1-1：widevine 整体框架图

DRM 框架与实现无关，可在方案特定的 DRM 插件中提取特定 DRM 方案实现的详情。DRM 框架包括可执行以下操作的简单 API: 处理复杂的 DRM 操作，获得许可，配置设备，将 DRM 内容与其许可相关联，以及最终解密 DRM 内容。

Android DRM 是在以下两个架构层中实现的：

• DRM framework API: 通过 Android 应用框架提供给应用。

• 本机代码 DRM 框架：为 DRM 插件（代理）提供接口，以便处理各种 DRM 方案的版权管理和解密工作。

1.2.2 工作原理

Android 平台提供了一个可扩展的 DRM 框架，支持应用根据与受版权保护的内容关联的许可限制条件来管理这些内容。DRM 框架支持多种 DRM 方案；设备具体支持哪些 DRM 方案由设备制造商决定。DRM 框架为应用开发者提供了一个统一接口，并隐藏了 DRM 操作的复杂性。DRM 框架为受保护和不受保护的内容提供了一致的操作模式。DRM 方案可以定义复杂的许可元数据使用模型。DRM 框架提供了 DRM 内容与许可之间的关联，并处理权限管理。这样可以将媒体播放器从受 DRM 保护或不受保护的内容中提取出来。

DRM 插件会与 Android DRM 框架集成在一起，并可使用受硬件支持的保护功能来确保付费内容和用户凭据的安全。

DRM 插件提供的内容保护功能取决于底层硬件平台的安全和内容保护功能。设备的硬件功能应包括硬件安全启动，可建立加密密钥的安全和保护功能的信任链。设备的内容保护功能应包括设备内加密帧的保护和通过可信输出保护机制实现的内容保护。并非所有硬件平台都支持上述所有的安全和内容保护功能。安全功能绝不会在堆栈的单个位置实现，而是依赖于硬件、软件和服务的集成。将硬件安全功能、可信启动机制以及用于处理安全功能的隔离安全操作系统组合使用是保障安全设备的关键。

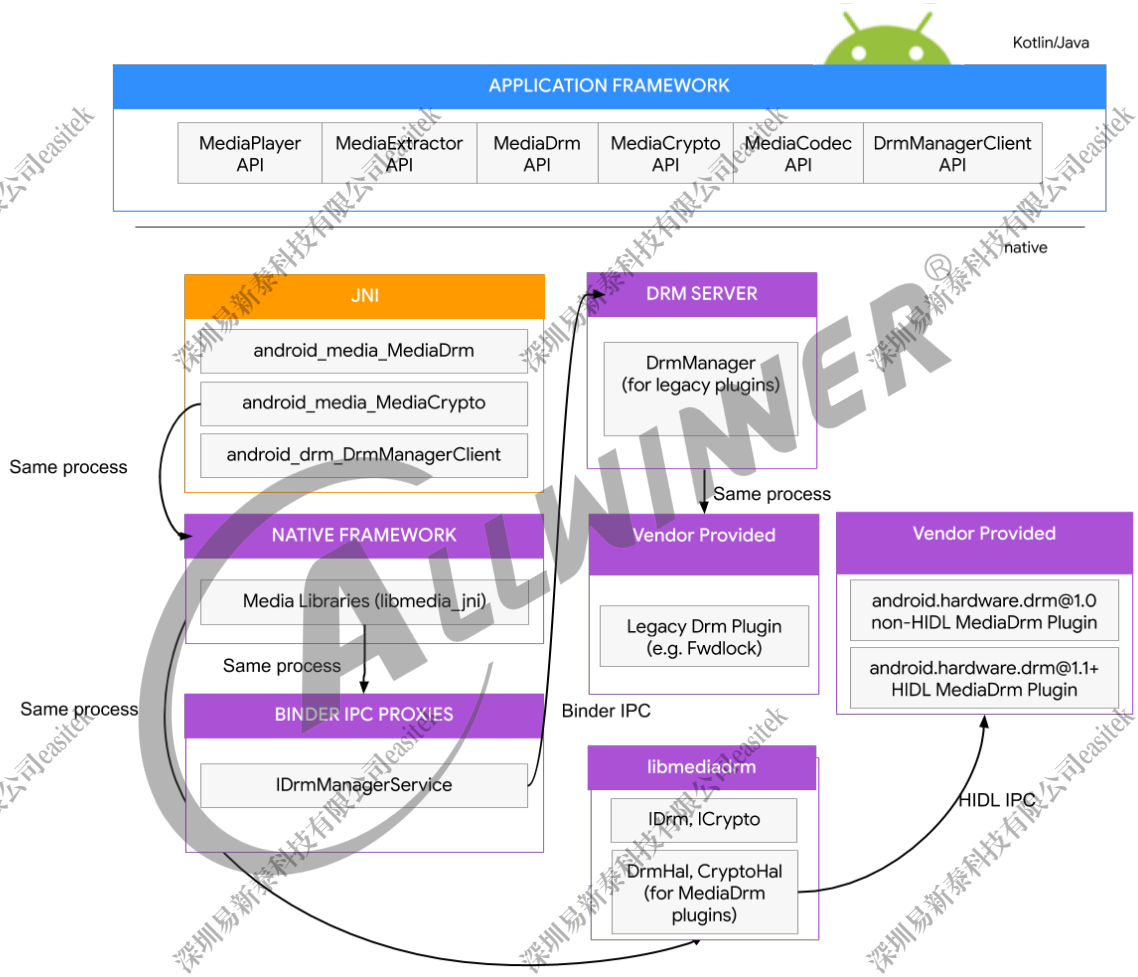

图 1-2: drm 插件图

对于 widevine 插件，在 Android 上同样是基于这个架构来进行：

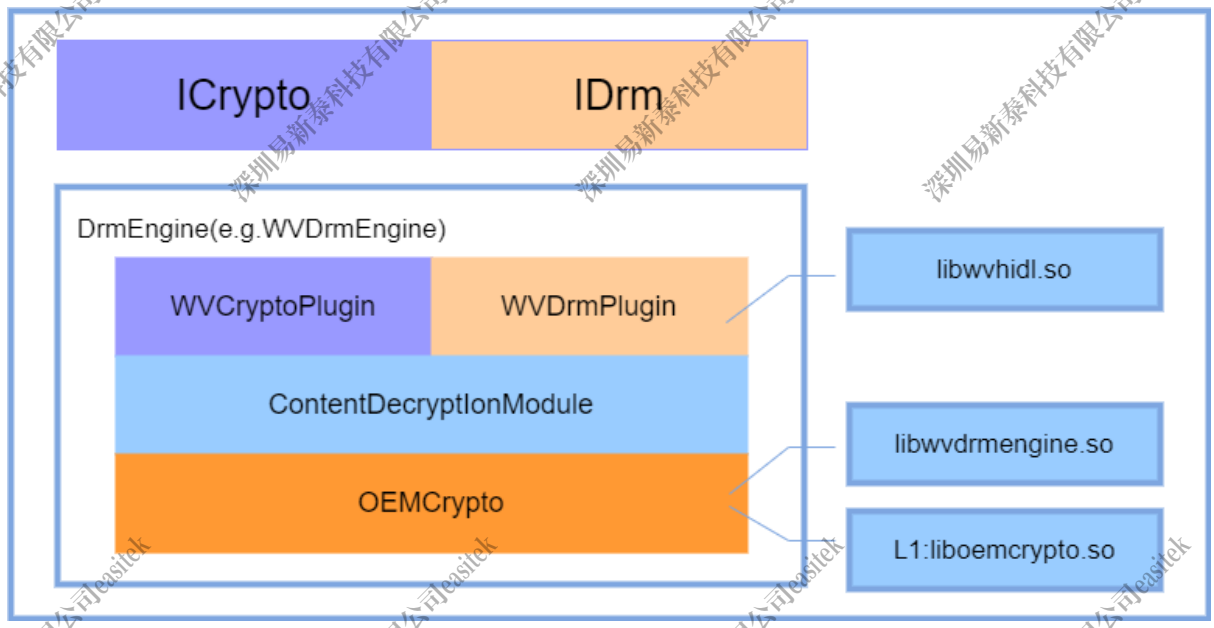

图 1-3: widevine 插件图

步骤 1：在播放前，WVDrmPlugin 向 DRM Server 申请 license，并获得认证通过。

步骤 2：媒体服务（MediaCodec) 向内容服务器下载加密内容，经过 MediaExtractor 解析将加密内容解析出来，然后调用 WVCryptoPlugin 进行解密，最后送码流到解码器进行解码播放。

1.2.3 安全等级

内容保护需要依赖于平台的安全能力，一个平台的安全能力需要安全硬件的支持，但并非所有的设备都具备该技术所需的硬件设备，故 widevine 分为了三个安全等级：

表 1-1: 安全等级表

安全等级	安全启动	widevine keybox 装配	安全硬件 TRUST-ZONE	KEYBOX 及视密钥处理	频硬件视频路径
Level 1	是	工厂安装	是	密钥不以明文暴露给 CPU	视频流通过硬件保护，输出在 TEE 中
Level 2	是	工厂安装	是	密钥不以明文暴露给 CPU	解码器直接获取明文视频流
Level 3	是	工厂安装	是	密钥以明文暴露给 CPU	解码器直接获取明文视频流

2 开发流程

2.1 Widevine L3 开发流程

widevineL3 由于不需要预装 Keybox，故 OEM 厂商不需要向 Google 申请 keybox，使用全志的 SDK，一般默认提供 widevineL3 的功能支持。

全志的 SDK 中，widevine 相关的库文件放在路径：*$android_root/hardware/aw/widevine* 下，如下：

```
├── android.hardware.drm@1.4-service-lazy.widevine
├── android.hardware.drm@1.4-service-lazy.widevine.rc
├── android.hardware.drm@1.4-service.widevine
├── android.hardware.drm@1.4-service.widevine.rc
├── Android.mk
├── CleanSpec.mk
├── demo
│   ├── ExoPlayerDemo.apk
│   └── readme.txt
├── lib32
│   ├── libvtswidevine.so
│   ├── libwvdrmengine.so
│   ├── libwvhidl.so
│   └── secure
│       └── liboemcrypto.so
├── manifest_android.hardware.drm@1.4-service.widevine.xml
├── nativetest
└── run_all_unit_tests_in_devies.sh

6 directories, 56 files
```

在使用 widevine L3 时，需要检查设备的配置选项，配置路径：*$/android/device/softwinner/apollo/common/secure/config.mk*, 配置为：

```
BOARD_WIDEVINE_OEMCRYPTO_LEVEL := 3
```

2.2 Widevine L1 开发流程

widevine L1 需要预装 keybox，并且需要安全硬件支持，流程较复杂。

步骤 1：OEM 厂商向谷歌签订协议

步骤 2：OEM 厂商获得全志的 SDK 及安全相关的 patch

步骤 3：OEM 厂商向谷歌申请购买 keybox

步骤 4：OEM 厂商在拿到 kebox 和全志 sdk 后，与全志集成开发，并根据生产环境定制生产工具装备 keybox。

步骤 5：完成测试，发布软件

步骤 6：OEM 产线生产和出货。

2.3 环境配置

1. Widevine L1 配置

设备配置 *$/android/device/softwinner/apollo/common/secure/config.mk*,

```
# drm config
BOARD_WIDEVINE_OEMCRYPTO_LEVEL := 1
```

widevine L1 需要安全固件，安全启动支持。

2. 安全 buffer 大小设置，默认安全 buffer 的大小只开了 80M，如果要播放大分辨率的视频，需要将此 buffer 增大，配置路径：*$/android/longan/device/config/chips/h618/configs/p2/sys_config.fex*:

```
[secure]
dram_region_mbytes = 80
drm_region_mbytes = 0
drm_region_start_mbytes = 0
```

3. mediacodec 配置，一般全志的 SDK 都是已经配置好了，可以确认 media_codec.xml（路径：*android/device/softwinner/apollo/common/media/codec*) 是否有配置 secure 解码器。
4. 镜像打包与烧写，需要安全固件（略）

2.4 keybox

widevineL1 要求在工厂生产时装备好 Keybox，如果没有装备 Keybox，就无法向 widevine license 服务器获取权限，就无法播放 widevine 视频。

1. keybox 定义

widevine Keybox 包含了一个唯一的 device ID，device key，加密密钥数据以及两个用于校验 Keybox 有效性的字段：常数字段及 CRC：

Field	Description	Size(bytes)
Device ID	C character string identifying the devide, null terminated.	32
Device Key	128 bit AES key assigned to device, generted by Widevine.	16
Key Data	Encrypted data	72
Magic	Constant code used to recognize a vaild keybox:"kbox" (0x6b626f78)	4
CRC	CRC-32-IEEE 802.3 validates integrity of the key data field	4
	Total Size	**128**

图 2-1: keybox 结构图

2. keybox 获取请求及传送协议

下图为获取的流程图：

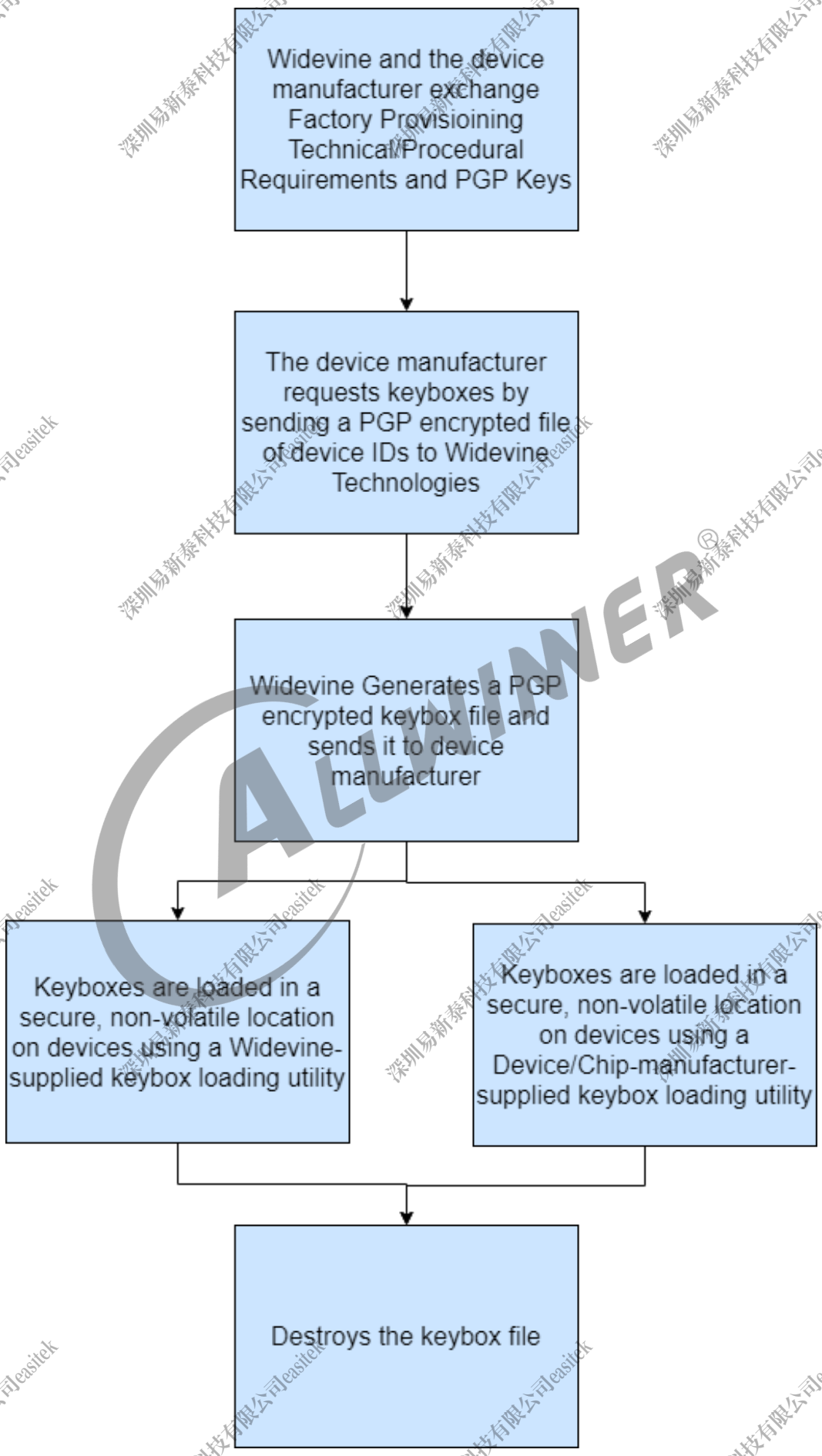

图 2-2: keybox 获取流程图

所有在设备厂商与 Widevine 之间的传递的关于 keybox 请求及相关文件必须通过 PGP 技术进行加密。

3. keybox 请求 Email 格式

厂商提供包含以下 keybox 请求信息：

• 请求信息正文：

1. Device Manufacturer: 公司名字
2. Device Model：产品型号
3. Number of keyboxes: 请求的 keybox 数量
4. Date：格式 mmddyyyy
5. Contact Email：有效的 email 地址

   例如，XYZ 公司在 12/25/2009 为设备型号为 BD1234 的产品请求 2 个 keybox 将在 emali 中包含如下的信息:

   1. Device Manufacturer:XYZ
   2. Device Model：BD1234
   3. Number of keyboxes:2
   4. Date:12252009
   5. Contact Email:contact@xyz.com

• Devices IDs 文件：

包含 Device ID 信息的文件必须用 PGP 进行加密并附在请求 email 中。文件名的格式为 MFGR_MODEL_DATE#OFIDS.ids。文件中必须是每个 Device id 一行，每个 device id 需要包含一个字母数字形式的字符串。最大的 device id 长度是 31 个字节。

例如，厂商 XYZ 将附上一个名为 XYZ_BD1234_12252009_2.ids，并使用设备序列号作为 device id 的一部分：

（1）XYZ_BD1234_808KVJH008324

（2）XYZ_BD1234_808KVJH008325

4. keybox 回复文件

在设备厂商请求了 keybox 之后，Widevine 会产生相应数量的 keybox 并置于一个 XML 格式的文件中。这个文件的文件名格式为 MFGR_MODEL_DATE_#OFIDS.keybox，例如 XYZ_BD1234_12252009_2.keybox，keybox 文件通过 PGP 加密后回复给厂商。

5. keybox xml 文件格式

keybox xml 文件格式如下：

```
<?xml version="1.0"?>
<Widevine>
<NumberOfKeyboxes>2</NumberOfKeyboxes>
<Keybox
DeviceID="mfg_mod123_0000001"><Key>c5f5cf3c2cb2ce175f2f5337a2f8f8ab</Key>
<ID>9d56e4931762b52aa21e4e590df477b5c81c683e0579f041ffa21f875c4c5e4a1cd4c2331
e27e3f4a49352fb432557336f63b1cb62549fddc9224b84d0c0364c827365fc217d9cb0</ID>
<Magic>6b626f78</Magic>
<CRC>0b11b841</CRC>
</Keybox>
<Keybox
DeviceID="mfg_mod123_0000002"><Key>73e38eb4f313e4fce8a5ab547cc7e2c0</Key>
<ID>215a40a9d13da3a9648335081a182869cbe78f607ce3ceb7506f351a22f411ae3f324ab5f
5bfb7c542ffcd38ec09438e7f92855149b02921463153c441332d7a21f875c4c5e4a1cd </ID>
<Magic>6b626f78</Magic>
<CRC>2b4c5e9f</CRC>
</Keybox>
</Widevine>
```

图 2-3: keybox 实例图

3 烧写 keybox

OEM 厂商从谷歌获得明文的 Keybox.xml 文件之后需要进行工厂烧写。对于工厂烧写 keybox，全志给出以下方案：

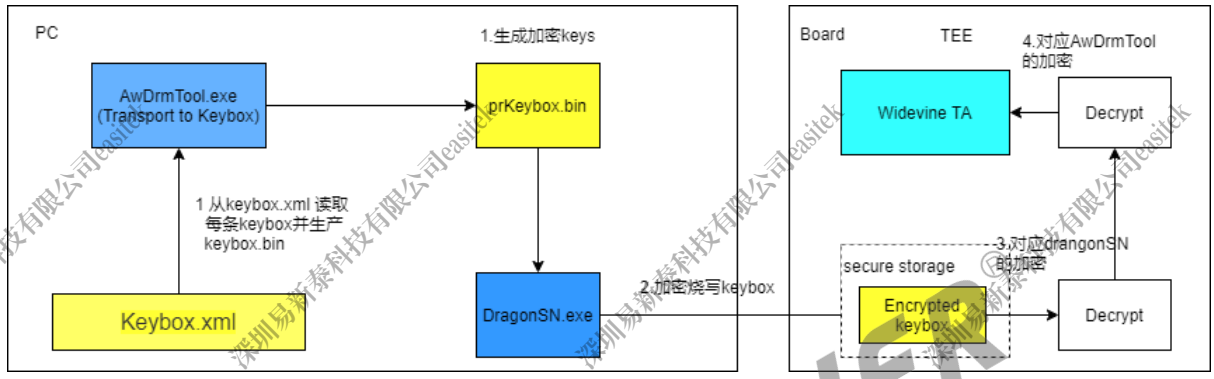

图 3-1：keybox 运行逻辑图

步骤 1：由研发人员使用 windows 工具 AwDrmTool.exe，从 Keybox.xml 里提取 keybox，注意生成的 keybox 应该要经过加密的，加密密钥是固定在软件里，用户不能更改，因为这个是对应 secure os 里的解密密钥。

工具的配置如图所示，生成的 keybox 命名为 wkey_N.bin，N 表示第几个：

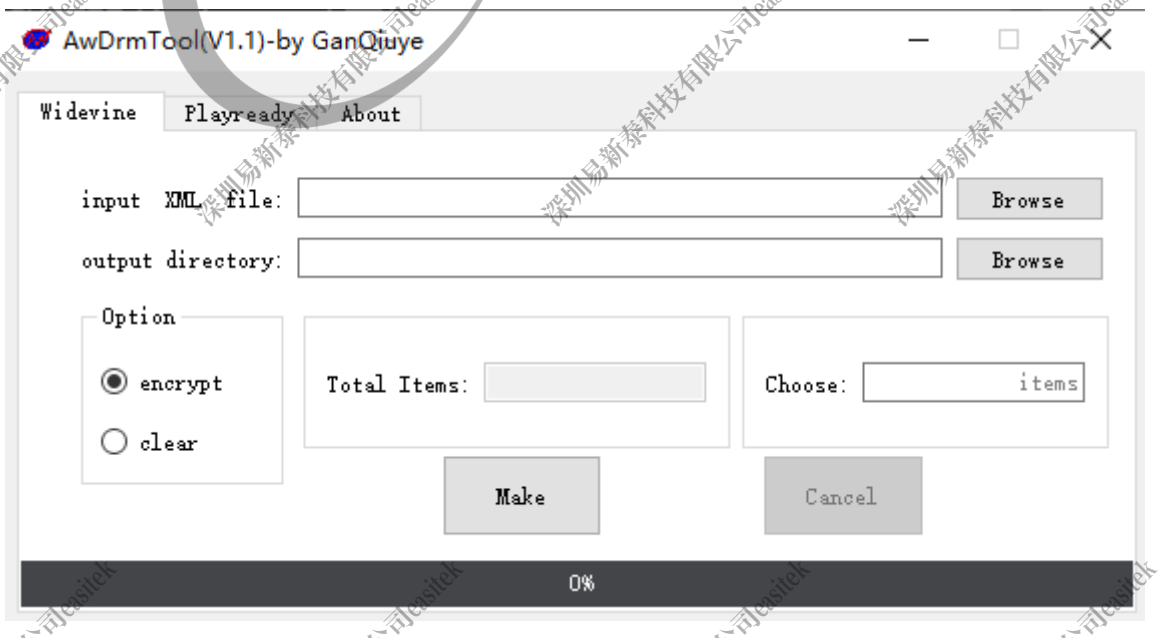

图 3-2：keybox 生成工具

警告

注意：Keybox.xml 只允许在实验室可见，产线只能拿到 wkey_N.bin。

步骤 2：使用 Dragon.exe 进行烧写 keybox.bin

1. 所烧的固件，确保 sdk device 中的配置：*android/device/softwinner/apollo/common/system/env.cfg* 的 keybox 配置带有 *widevine* 字样

```
keybox_list=widevine
```

2. 配置 DragonSN，添加 widevine 相关的配置

图 3-3: DragonSN 配置图 1

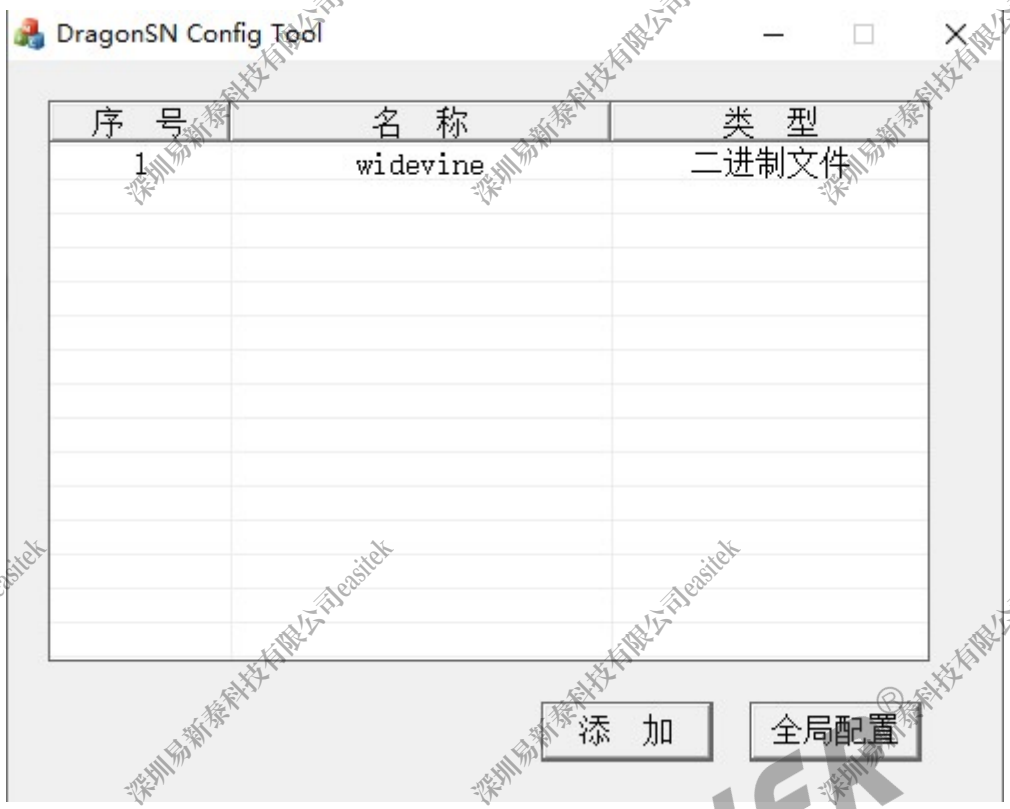

图 3-4: DragonSN 配置图 2

3. DragonSN 的 widevine 全局配置

图 3-5: DragonSN 配置图 3

4. DrangonSN 的全局配置中有些参数没有写进界面中，故需要去到工具的安装文件夹中找到 global.ini，修改 widevine 的配置：

```
[global]
huk=0
setflag=0
setdata=0
burnmode=0
db_server=
db_port=0
db_user=
db_pwd=
db_database=
db_type=mssql
db_primary_key=
db_default_table=
solution_code=
db_used_column=
db_used_value=
db_valid_value=
root=\ULI\factory
[widevine]
key_name=widevine
key_type=1
key_burn=1
key_replace=1
key_crype=1
partition_check=0
key_itemburnmode=2
data_type=key
data_source=bins
total=0
remain=0
increase_mode=
db_primary_key=
db_table=
value=
addition=
match=
verify_key=
```

图 3-6: DragonSN 配置图 4

5. 运行 DragonSN，选择 keybox bin 所在目录进行烧写。连上 adb，设备上电开启，点击烧写按钮进行烧写。如果烧写按钮没有使能，就需要进入 bootloader 中，使用串口工具输入命令：*reboot bootloader*，然后输入 *uburn* 就可以使能烧写按钮。

警告

注意：DragonSN 每烧一个 Keybox，就会将对应的 keybox.bin 移到一个标记为已使用的文件夹里。根据谷歌的要求，已烧写了的 Keybox 需要妥善移除。

4 测试

4.1 GTS 测试

4.2 Exoplayer 测试

4.3 其他播放器测试

著作权声明

本文档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本文档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

商标声明


、 
、全志科技、  （不完全列举）均为珠海全志科技股份有限公司的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的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所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。