设备证明集成开发指南 标准设备

文档版本 02
发布日期 2023-10-10

1 修改记录

表1 修改记录

文档版本	修改日期	修改说明
01	2023-02-03	发布设备证明集成开发指南
02	2023-10-10	登记信息描述规范、软证书格式调整

2 名词解释

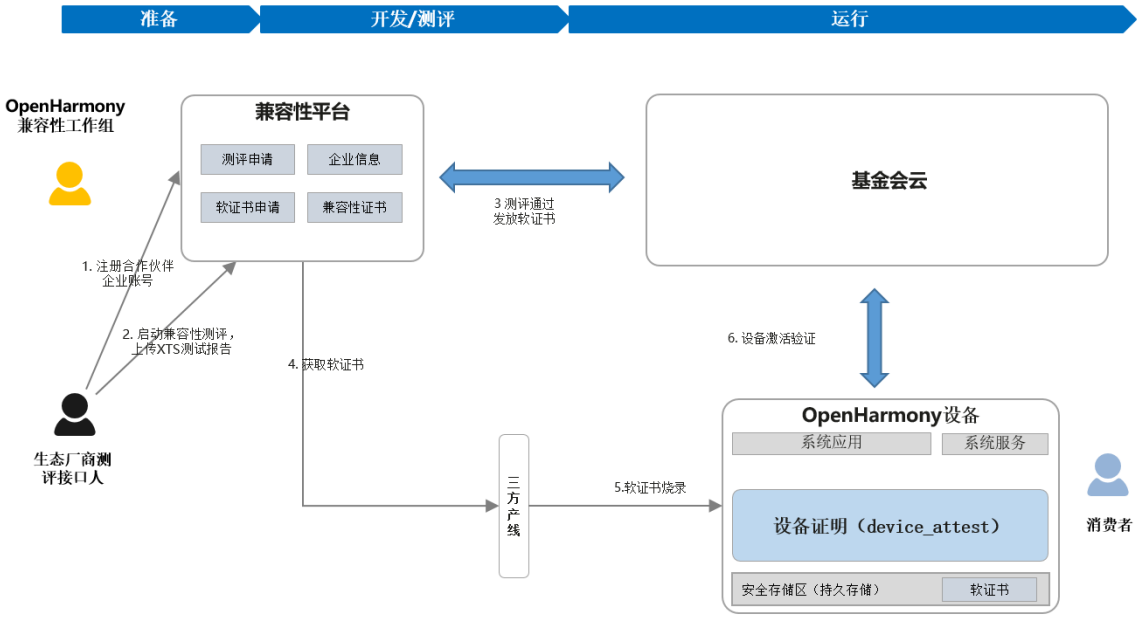
表2 名词解释

名词	解释
伙伴	申请OpenHarmony兼容性测评的企业，下文统称为“伙伴”。需要集成设备证明模块。
设备证明	设备证明是一个系统服务, 是OpenHarmony Compatibility Agreement约定需要设备厂商在产品中集成的部件，用于支撑生态伙伴完成产品的兼容性测试。
授权验证	授权验证包括设备侧的设备证明与基金会云侧的校验服务。
软证书	伙伴从OpenHarmony兼容性平台获取，由平台分配的设备凭据，每台设备一个，标识设备身份。需存储在安全分区，在恢复出厂设置、镜像升级时也不能被清除。
manuKey	伙伴从OpenHarmony兼容性平台获取的秘钥。用于对产品中相关数据进行加密保护。为了保证多产品的兼容性，manuKey在所有产品的生命周期内都应保持不变。
productId	伙伴从OpenHarmony兼容性平台申请兼容性测评时，平台为测评产品分配的唯一产品标识。productId在产品生命周期内需保证始终不变。
productKey	伙伴从OpenHarmony兼容性平台申请兼容性测评时，平台为测评产品分配的唯一产品秘钥。其与productId一一对应，用于对产品级的数据进行加密保护。同样需保证在产品生命周期内始终保持不变。

3 整体介绍

3.1 授权验证功能

授权验证包括设备侧集成的设备证明模块与基金会云侧的校验服务。通过端云结合的方式验证当前设备是否为基金会通过兼容性测评的设备。



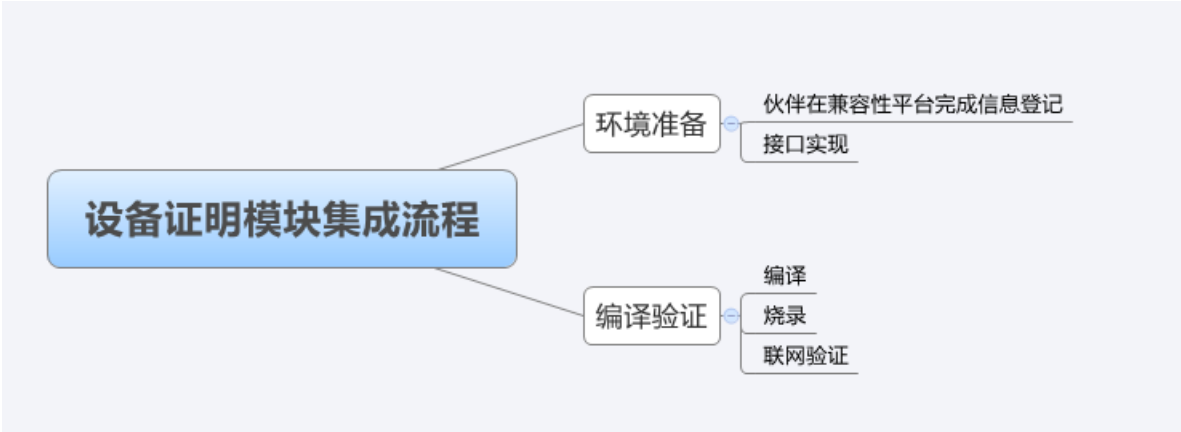
3.2 依赖库

设备证明模块的集成依赖相关库如下表：

表3 依赖相关库

库名称	版本号（包含及以上）	功能描述	仓库路径
mbedtls	2.16.11	供嵌入式设备使用的一个TLS协议的轻量级实现库。	third_party\mbedtls
OpenSSL	1.1.1	传输层安全（TLS）协议（包括SSLv3）以及通用密码库。	third_party\openssl
cJSON	1.7.15	JSON 文件解析库。	third_party\cJSON\third_party
libsec	1.1.10	安全函数库。	bounds_checking_function
parameter	OpenHarmony 1.0+	获取设备信息的系统接口。	base\startup\init

3.3 设备证明模块集成流程



4 环境准备

4.1 伙伴完成信息登记

伙伴需要在OpenHarmony兼容性平台注册关于产品设备的一系列基础信息，如：公司简称（英文）、品牌英文名称、设备型号等。
设备联网后，设备证明模块读取设备信息并上报基金会云，基金会云进行校验验证。因此需要伙伴提前在 OpenHarmony 兼容性平台官网上完成产品的信息登记，分如下两个步骤：
1）伙伴在OpenHarmony兼容性平台上完成设备信息登记。
2）伙伴将OpenHarmony兼容性平台上登记的设备信息写入设备。

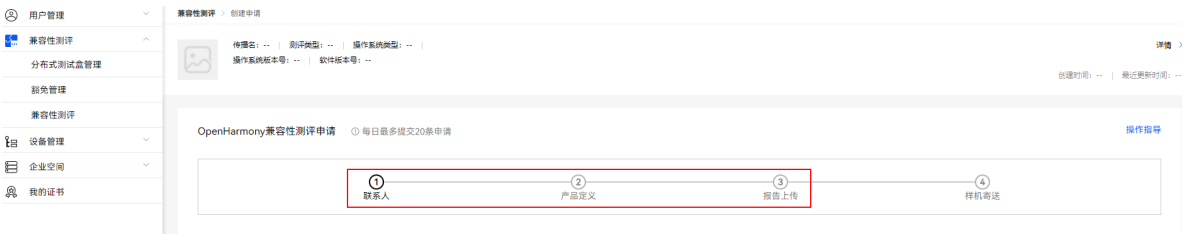
4.1.1 伙伴在OpenHarmony上完成信息登记

伙伴需要在OpenHarmony兼容性平台上注册相关设备数据，请按照官网上的注册流程完成。

登录兼容性平台选择“兼容性测评” > “创建申请”

1）在申请页面完成“联系人”与“产品定义”两步填写后，点击“下一步”后退出，可以申请软证书进行设备证明模块调试，但是端云校验会失败；

2）在申请页面完成“联系人”与“产品定义”、“报告上传”三步填写后，点击“下一步”后退出，可以申请软证书进行设备证明模块调试，集成正常前提下端云校验成功；



4.1.2 伙伴将登记信息写入设备

针对伙伴在OpenHarmony兼容性平台上登记的信息，版本包提供了相关接口供伙伴进行填写。
在调用设备证明模块对外接口时，会将伙伴填写的值上报给基金会云，基金会云会将4.1.1 章节注册的信息与设备上报的信息进行对比校验。
设备证明模块依赖部分设备信息，需要伙伴适配修改，请查看表4和表5。

表4 设备OS信息

设备信息	配置参数	备注
发布类型	const.ohos.releasetype=Release	使用默认值
api版本	const.ohos.apiversion=9	使用默认值
安全补丁标签	const.ohos.version.security_patch=2023/03/01	2023/03/01需要替换成真实值
操作系统及版本号	const.ohos.fullname=OpenHarmony-3.2.11.9	使用默认值

设备OS信息配置文件路径：

```
base/startup/init/services/etc/param/ohos_const/ohos.para
```

表5 设备产品信息

设备信息	配置参数	备注
企业简称（英文）	const.product.manufacturer=default	default 需要替换成设备真实值
品牌英文名	const.product.brand=default	default 需要替换成设备真实值
设备型号	const.product.model=default	default 需要替换成设备真实值
用户可见的软件版本号	const.product.software.version="OpenHarmony 1.0.1"	"OpenHarmony 1.0.1" 需要替换成设备真实值
版本id	-	不需要伙伴修改，系统自动生成
版本Hash	const.ohos.buildroothash=default	如果使用默认值，兼容性平台的“版本Hash”需要保持一致default

设备产品信息配置文件路径：

```
base/startup/init/services/etc/param/ohos.para
```

注：版本id，是系统根据拼接规则自动生成的。拼接规则如下：

> deviceType/manufacture/brand/productSeries/OSFullName/productModel/softwareModel/OHOS_SDK_API_VERSION/incrementalVersion/buildType

版本 id获取方法:
方法 1)输入shell命令 begetctl dump api获取设备信息

```
# begetctl dump api
Begin dump syspara
======================
DeviceType:default
Manufacture:default
Brand:default
MarketName:OpenHarmony 3.2
ProductSeries:default
ProductModel:ohos
SoftwareModel:default
HardwareModel:default
Serial:7001005458323933328a25da7cd23900
OSFullName:OpenHarmony-3.2.10.11(Beta5)
DisplayVersion:OpenHarmony 3.2.10.11
BootloaderVersion:bootloader
GetSecurityPatchTag:2023-03-01
AbiList:default
IncrementalVersion:default
VersionId:default/default/default/default/OpenHarmony-3.2.10.11(Beta5)/ohos/default/9/default/default
BuildType:default
BuildUser:default
BuildHost:default
BuildTime:default
BuildRootHash:default
GetOsReleaseType:Beta5
GetHardwareProfile:default
FirstApiVersion:1
GetSerial:7001005458323933328a25da7cd23900
acl serial:1234567890
GetDevUdid:D8CA3D6BB429296B2C9049D661EBB7B2049CCA0C949B51C33E8E8639B01C5F6E
Acl devUdid:
Version:3.2.10.11
GetSdkApiVersion:9
GetSystemCommitId:788
======================
End dump syspara
```

方法 2) 使用标准接口获取GetVersionId

4.2 依赖接口适配

为了屏蔽不同硬件实现差异，设备证明模块定义了软证书相关API，由伙伴具体适配实现，需要伙伴替换接口定义的归属文件，接口定义如表6。
标准系统的接口定义的归属文件：

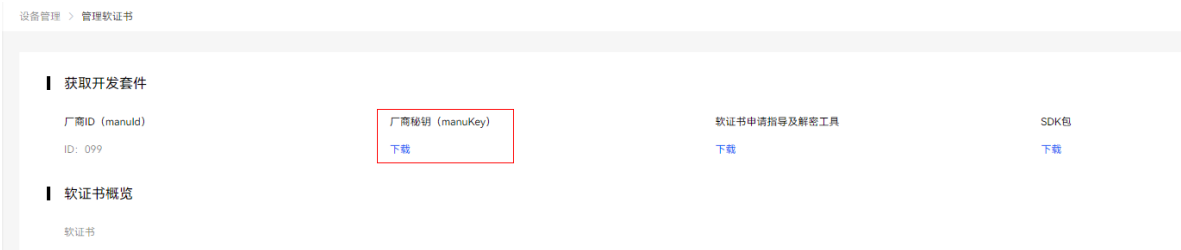
```
..\test\xts\device_attest\services\oem_adapter\src\device_attest_oem_adapter.c
```

表6 接口定义

功能	接口定义	参数定义	返回值	适配说明
读取manuKey	int32_t OEMGetManufacturekey(char* manufacturekey, const uint32_t len)	manufacturekey：秘钥存储内存 len：内存长度	0：成功 -1：失败	替换manufacturekeyBuf内容
读取ProductId	int32_t OEMGetProductId(char* productId, const uint32_t len)	productId：产品型号标识 len：存储空间长度	0：成功 -1：失败	替换productIdBuf内容
读取软证书	int32_t OEMReadToken(char *token, const uint32_t len)	token：存储软证书的空间 len：存储软证书的长度	0：成功 非0：失败	自行实现相关功能
写入软证书	int32_t OEMWriteToken(const char *token, const uint32_t len)	token：存储软证书的空间 len：存储软证书的长度	0：成功 -1：失败	自行实现相关功能，需要存储在设备安全分区，设备重启、初始化都不会擦除的分区

1）OEMGetManufacturekey

manuKey，即厂商秘钥，是和软证书配套，生成AES秘钥的参数，具体从OpenHarmony兼容性平台下载。进入OpenHarmony兼容性平台后，选择“设备管理” > “管理软证书”，点击“厂商秘钥（manuKey）”下方的“下载”。



下载文件内容进行ASCII码转16进制，例如可以直接使用Notepad++等自带的转换功能（“插件” > “Converter” > "ASCII -> HEX"）。找到接口定义的归属文件，将16进制的文件内容赋值给manufacturekeyBuf[]。

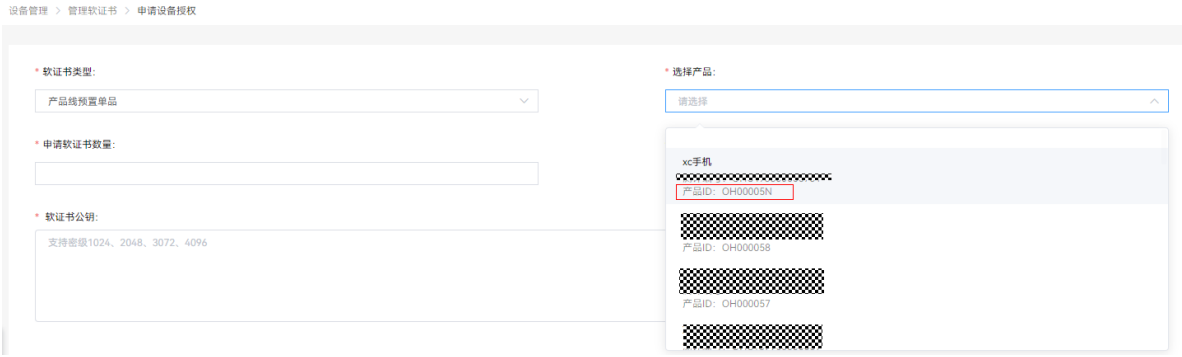
```
int32_t OEMGetManufacturekey(char* manufacturekey, const uint32_t len)
{
    if ((manufacturekey == NULL) || (len == 0)) {
        return DEVICE_ATTEST_OEM_ERR;
    }

    const char manufacturekeyBuf[] = {
        0x13, [redacted] 0x42,
        0x3F, [redacted] 0x3F,
        0x57, [redacted] 0x47
    };
    uint32_t manufacturekeyBufLen = sizeof(manufacturekeyBuf);
    if (len < manufacturekeyBufLen) {
        return DEVICE_ATTEST_OEM_ERR;
    }

    int32_t ret = memcpy_s(manufacturekey, len, manufacturekeyBuf, manufacturekeyBufLen);
    return ret;
}
```

2）OEMGetProductId

ProductId，即产品ID，是和软证书配套，生成AES秘钥的参数，具体可以在OpenHarmony平台查看。进入OpenHarmony兼容性平台后，选择“设备管理” > “管理软证书”，点击“申请设备授权”，在“选择产品”的下拉栏中，即可查看和选择对应的产品ID。（如果没有相关信息，证明4.1.1章节未录入设备信息，请完善后再操作）



查看完产品ID后，需要在接口定义的归属文件，接口OEMGetProductId里，替换productIdBuf[]的值。

```
int32_t OEMGetProductId(char* productId, const uint32_t len)
{
    if ((productId == NULL) || (len == 0)) {
        return DEVICE_ATTEST_OEM_ERR;
    }

    const char productIdBuf[] = "OH00004O";

    uint32_t productIdLen = strlen(productIdBuf);
    if (len < productIdLen) {
        return DEVICE_ATTEST_OEM_ERR;
    }

    int32_t ret = memcpy_s(productId, len, productIdBuf, productIdLen);
    return ret;
}
```

3）OEMReadToken和OEMWriteToken

token，即软证书。厂商需要实现软证书读和写接口，把软证书写在设备的安全分区，设备重启、初始化都不会擦除的分区。

5 对外接口

无相关使用需求，可以跳过该章节

对外接口所在路径：

```
..\test\xts\device_attest\interfaces\innerkits\native_cpp\include\devattest_client.h
```

表7 对外接口

native接口	描述
int32_t GetAttestStatus(AttestResultInfo* attestResultInfo);	获取授权验证结果

设备证明部件开机自启动，网络连接成功后，会进入设备证明部件主流程。通过调用GetAttestStatus接口，获取授权验证结果。

JS接口	描述
function getAttestStatus(callback: AsyncCallback) : void	获取设备授权验证结果(异步接口)
function getAttestStatus() : Promise;	获取设备授权验证结果(异步接口)
function getAttestStatusSync() : AttestResultInfo;	获取设备授权验证结果(同步接口)

优先使用异步接口，异步接口同时支持callback和Promise两种方式，使用哪种方式由开发者决定。

6 集成与验证

6.1 编译

设备证明模块属于XTS子系统，OpenHarmony3.2及以后版本需要集成。

产品代码配置表中增加设备证明部件，如下：

```
{
  "subsystem": "xts",
  "components": [
    {
      "component": "device_attest",
      "features": []
    }
  ]
},
```

举例以rk3568编译指令如下：

```
./build.sh --product-name=rk3568 system_size=standard
```

编译成功后会在out/rk3568/packages/phone/system/lib路径下生成libdevattest_core.z.so、libdevattest_permission.z.so、libdevattest_sdk.z.so、libdevattest_service.z.so、libdevice_attest_oem_adapter.z.so、libdeviceattest.z.so六个动态库。

6.2 烧录

伙伴在完成OpenHarmony的构建后，将编译出的镜像文件烧录到设备上，对开发的代码进行调试和验证。

6.3 验证阶段软证书导入

6.3.1 软证书申请指导及解密工具

在OpenHarmony兼容性平台，选择“设备管理” > “管理软证书”，点击“软证书申请指导及解密工具”下方的“下载”。参考指导，获取软证书。

6.3.2 软证书格式

具体软证书导入的格式，由伙伴对OEMReadToken和OEMWriteToken接口的具体实现决定。

设备证明模块代码实现默认软证书格式：

1）创建名为“tokenA”不带后缀名的文件

2）打开文件，输入“01020304”，并使用16进制转ASCII码功能（“插件” > “Converter” > "HEX -> ASCII"）

3）把获取的软证书拷贝至后面（下载的软证书文件，一行为一个软证书）

4）在软证书后输入“00000000”，并使用16进制转ASCII码功能（“插件” > “Converter” > "HEX -> ASCII"）

6.3.3 hdc指令导入

镜像烧录完成之后，使用hdc指令发送文件到设备安全区，具体路径由伙伴适配，需和device_attest_oem_adapter.c中的路径一致。（调测阶段可以手工推送软证书，商用设备需要产线完成预置）

举例：如果使用默认软证书格式，tokenA文件在E盘根路径下，使用默认设备证明模块依赖接口。

```
hdc_std file send E:\tokenA data/device_attest/tokenA
```

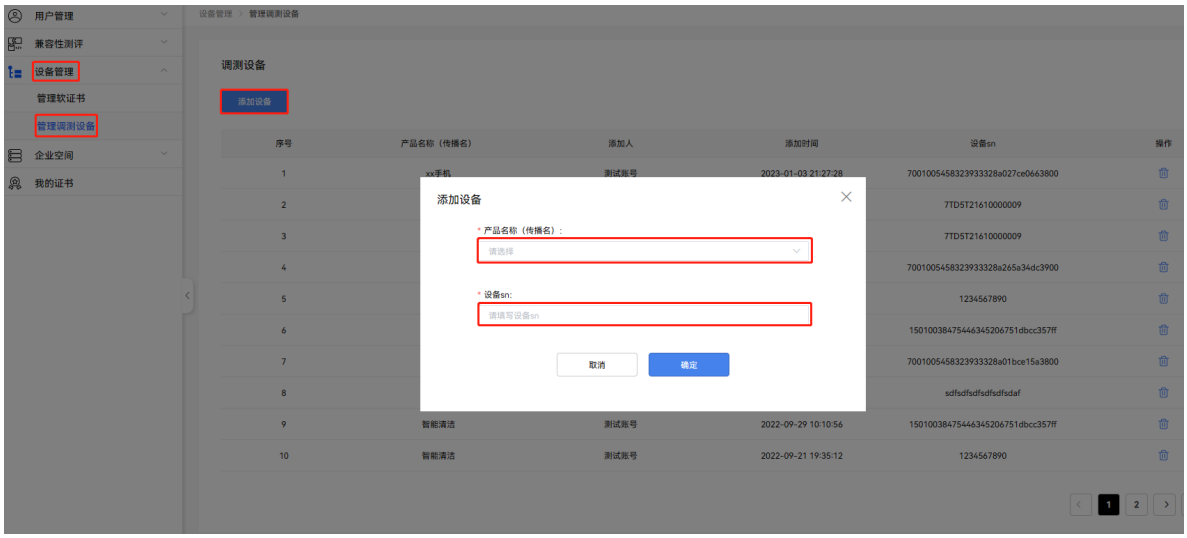
还需要对刚发送的tokenA文件，进行权限修改。使用hdc指令进入shell，输入以下指令。

```
chown device_attest data/device_attest/tokenA
```

6.4 调测

6.4.1 准备工作

伙伴在OpenHarmony兼容性平台，选择“设备管理” > “管理调测设备”，点击添加设备，选择刚注册产品的产品名称，填写调测设备的sn号，点击“确定”按钮。



6.4.2 结果验证

伙伴将设备完成烧录、软证书导入以及调测准备工作后，可通过联网对设备进行授权验证功能调测，通过访问查询接口5 对外接口，根据获取的值来查看授权验证结果。

结构体attestResultInfo中authResult 的值为0，则说明软证书端云校验成功。

结构体attestResultInfo中softWareResult 的值为0，则说明设备软件信息与4.1.2章节录入信息一致。

authResult 与softWareResult 字段均为0，则说明授权验证验证成功。

说明：

代码中提供sample方便设备调试，使用方法：

1）打开编译开关

test\xts\device_attest\build\devattestconfig.gni 修改 enable_attest_test_sample = true

2）版本编译

编译产物中会生成attesttestclient （\xts\device_attest目录下）可执行二进制文件，在设备端执行查看授权验证结果。

7 商用

调测验证成功后，则证明设备授权验证打通。

兼容性测评通过后，需要在兼容性平台批量申请软证书，进行产线烧录（每个设备一个软证书，不能重复使用）。

说明：只有在调测阶段需要在OpenHarmony兼容性平台录入调测设备sn，测评通过后如果想基于该设备验证正式环境，删除录入sn信息即可。