Linux 安全
开发指南

版本号: 1.3
发布日期: 2021.11.25

深圳市索智科技有限公司SuoZhi

版本历史

版本号	日期	制/修订人	内容描述
1.0	2020.11.10	AWA1551	1. 添加初始版本
1.1	2021.03.02	AWA1551	1. 兼容 4.9 版本 kernel
1.2	2021.05.13	AWA1551	1. 修正适用范围描述
1.3	2021.11.25	AWA1551	1. 根据 SDK 调整烧 key 开关的说明 2. 修正部分标题的格式错误

深圳市索智科技有限公司SuoZhi

版权所有 © 珠海全志科技股份有限公司。保留一切权利

目　录

1 前言 **1**
1.1 文档简介 . . . 1
1.2 目标读者 . . . 1
1.3 适用范围 . . . 1

2 安全系统基础 **2**
2.1 安全系统介绍 . . . 2
2.2 安全基础介绍 . . . 2
2.2.1 数据加密模型 . . . 2
2.2.2 加密算法 . . . 2
2.2.3 签名与证书 . . . 4
2.2.4 efuse . . . 6
2.3 TrustZone . . . 7
2.3.1 OPTEE . . . 7
2.4 相关术语 . . . 8

3 硬件安全模块 **9**

4 secure boot **10**
4.1 基本原理 . . . 10
4.1.1 基于证书的哈希校验 . . . 10
4.1.2 防回滚版本号校验 . . . 11
4.2 配置密钥 . . . 12
4.3 配置防回滚版本号 . . . 12
4.4 量产模式说明 . . . 13
4.5 生成安全固件 . . . 13
4.6 ROTPK 烧写 . . . 14
4.7 注意事项 . . . 15

5 secure os **16**
5.1 OPTEE 介绍 . . . 16
5.1.1 OPTEE OS . . . 16
5.1.2 OPTEE linux driver . . . 17
5.1.3 OPTEE client . . . 17
5.1.4 TA/CA . . . 17
5.2 TEE 运行环境配置 . . . 17
5.2.1 OPTEE OS . . . 18
5.2.2 OPTEE linux driver . . . 18
5.2.3 OPTEE client . . . 19
5.2.4 TEE 环境内存空间配置 . . . 20
5.3 TEE 运行环境的使用 . . . 21

6 TA/CA 开发指引 **22**

版权所有 © 珠海全志科技股份有限公司。保留一切权利

6.1 开发环境目录结构 . . . 22
6.2 编译 . . . 22
6.3 编译脚本使用说明 . . . 22
6.4 拷贝 . . . 22
6.5 运行 . . . 23
6.5.1 运行辅助 REE 与 TEE 通信的守护进程 . . . 23
6.5.2 运行 DEMO . . . 23
6.5.2.1 helloworld . . . 23
6.5.2.2 encrypt_file_storage . . . 24
6.5.2.3 base64-usage . . . 27
6.6 编译配置 . . . 28
6.6.1 TA . . . 28
6.6.2 CA . . . 29
6.6.3 TA 加密 . . . 30

7 密钥存储 31
7.1 efuse . . . 31
7.2 flash . . . 32
7.2.1 安全 key(secure storage) . . . 32
7.2.1.1 keybox . . . 32
7.2.2 私有 key(private storage) . . . 34

8 TEE 环境中数据的掉电保存 38
8.1 OPTEE Secure Storage 功能框架 . . . 38
8.2 文件操作流程 . . . 39
8.3 使用 OPTEE Secure Storage 为 REE 提供加密文件存储 . . . 39

9 参考资料 40

深圳市索智科技有限公司SuoZhi

版权所有 © 珠海全志科技股份有限公司。保留一切权利

插　图

2-1 加密交互过程 2
2-2 对称及非对称加密 3
2-3 摘要计算 3
2-4 数字签名 4
2-5 验证签名 5
2-6 证书结构 6
2-7 TurstZone 系统模型 7
4-1 固件验证过程 11
4-2 防回滚版本号配置文件 13
4-3 配置烧 key 属性 14
4-4 rotpk 烧录时小机端处理 14
5-1 OPTEE 总体架构 16
5-2 进入驱动配置 18
5-3 使能 linux TEE driver 19
5-4 使能 TEE driver 中 OP-TEE 的 driver 19
7-1 DragonSN 烧录安全 efuse 配置 31
7-2 DragonSN 烧录安全 key 配置 32
7-3 key 烧录到 keybox 的过程 33
7-4 key 从 keybox 中加载的过程 34
7-5 DragonSn 烧录私有 key 配置 35
7-6 格式化私有分区 36
7-7 私有 key 烧录路径 37
8-1 OPTEE Secure Storage 软件架构 38

深圳市索智科技有限公司SuoZhi

版权所有 © 珠海全志科技股份有限公司。保留一切权利

1 前言

1.1 文档简介

本文主要介绍了 Allwinner 安全方案的组成与功能。安全完整的方案基于 normal 方案扩展, 覆盖硬件安全、安全启动（Secure Boot）、安全系统（Secure OS）、安全应用（Trusted apps）等方面。本文从硬件安全、安全启动（Secure Boot）、安全系统（Secure OS）、安全应用的开发（TA/CA 开发指引）、固件密钥存储、安全系统在 Flash 上的加密保存几个方面进行介绍。

1.2 目标读者

Allwinner 软件平台的相关技术人员

1.3 适用范围

Allwinner 软件平台

Allwinner 使用 4.9 或更高版本的 kernel 的硬件平台

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2 安全系统基础

2.1 安全系统介绍

安全系统是基于硬件配合软件的安全解决方案。其主要目的是保障系统资源的完整性、保密性、可用性，从而为系统提供一个可信的运行环境。

2.2 安全基础介绍

2.2.1 数据加密模型

a. 明文 P。准备加密的文本，称为明文。

b. 密文 Y。加密后的文本，称为密文。

c. 加解密算法 E(D)。用于实现从明文到密文或从密文到明文的一种转换关系。

d. 密钥 K。密钥是加密和解密算法中的关键参数。

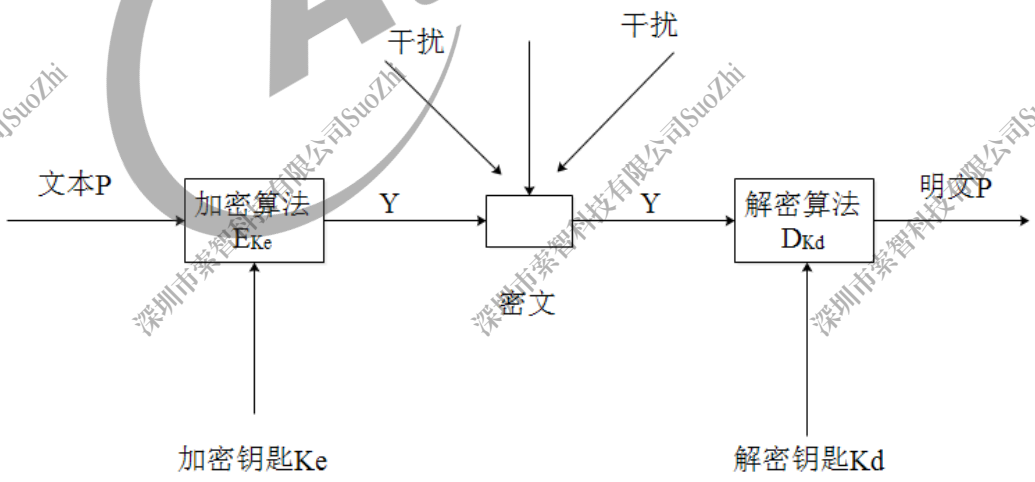

图 2-1: 加密交互过程

2.2.2 加密算法

a. 对称加密算法: 加密、解密用的是同一个密钥。比如 AES 算法。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

b. 非对称加密算法：加密、解密用的是不同的密钥，一个密钥“公开”，即公钥，另一个密钥持有，即私钥。其中一把用于加密，另一把用于解密。比如 RSA 算法。

c. 散列（hash）算法：一种摘要算法，把一笔任意长度的数据通过计算得到固定长度的输出，但不能通过这个输出得到原始计算的数据。

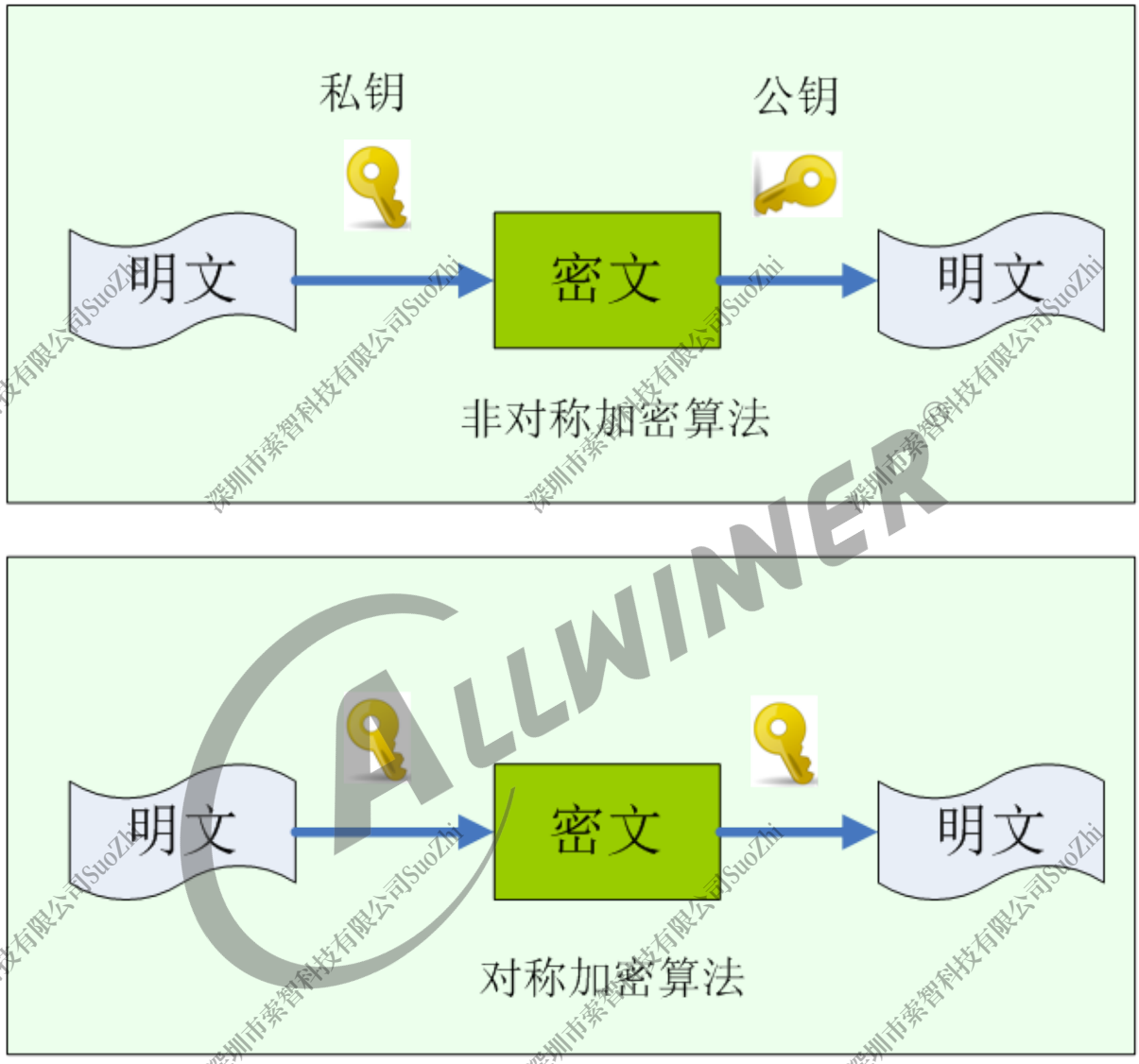

图 2-2：对称及非对称加密

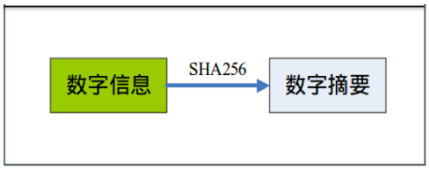

图 2-3：摘要计算

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.2.3 签名与证书

数字签名：数字签名是非对称密钥加密技术与数字摘要技术的应用。数字签名保证信息是由签名者自己签名发送的，签名者不能否认或难以否认；可保证信息自签发后到收到为止未曾作过任何修改，签发的文件是真实文件

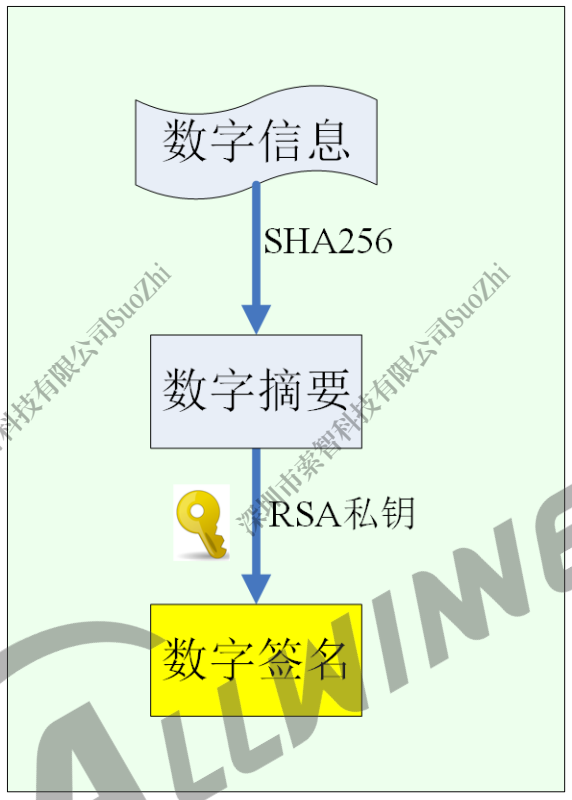

图 2-4：数字签名

版权所有 © 珠海全志科技股份有限公司。保留一切权利

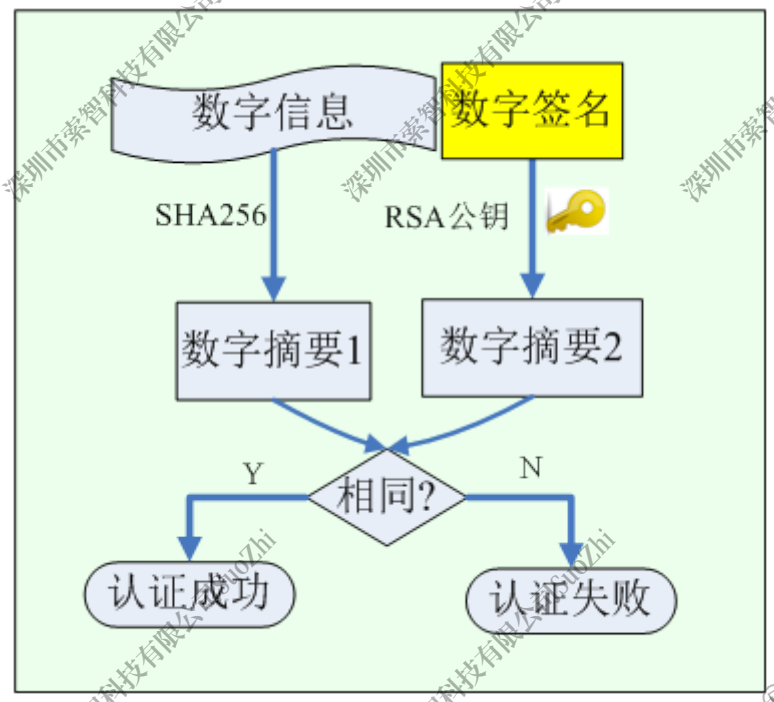

图 2-5: 验证签名

数字证书: 是一个经证书授权中心数字签名的包含公开密钥拥有者信息以及公开密钥的文件，是一种权威性的电子文档。

深圳市索智科技有限公司SuoZhi

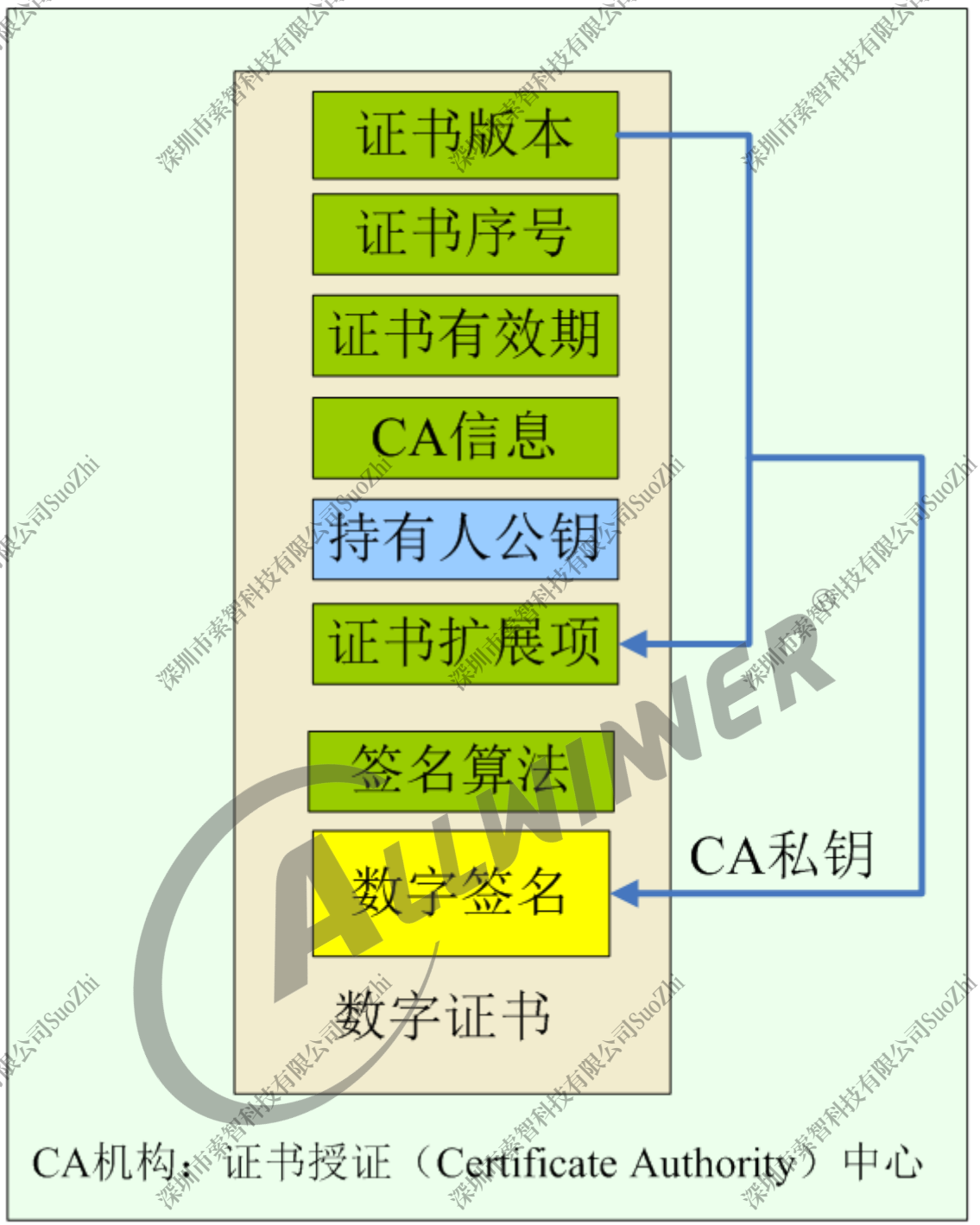

图 2-6: 证书结构

2.2.4 efuse

efuse: 一次性可编程熔丝技术。有些 SoC 集成了一个 efuse 电编程熔丝作为 OTP（One-Time Programmable, 一次性可编程）存储器。efuse 内部数据只能从 0 变成 1, 不能从 1 变成 0, 只能写入一次。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.3 TrustZone

TrustZone 是 ARM 提出的安全解决方案，旨在提供独立的安全操作系统及硬件虚拟化技术，提供可信的执行环境（Trust Execution Environment）。TrustZone 系统模型如下图所示。

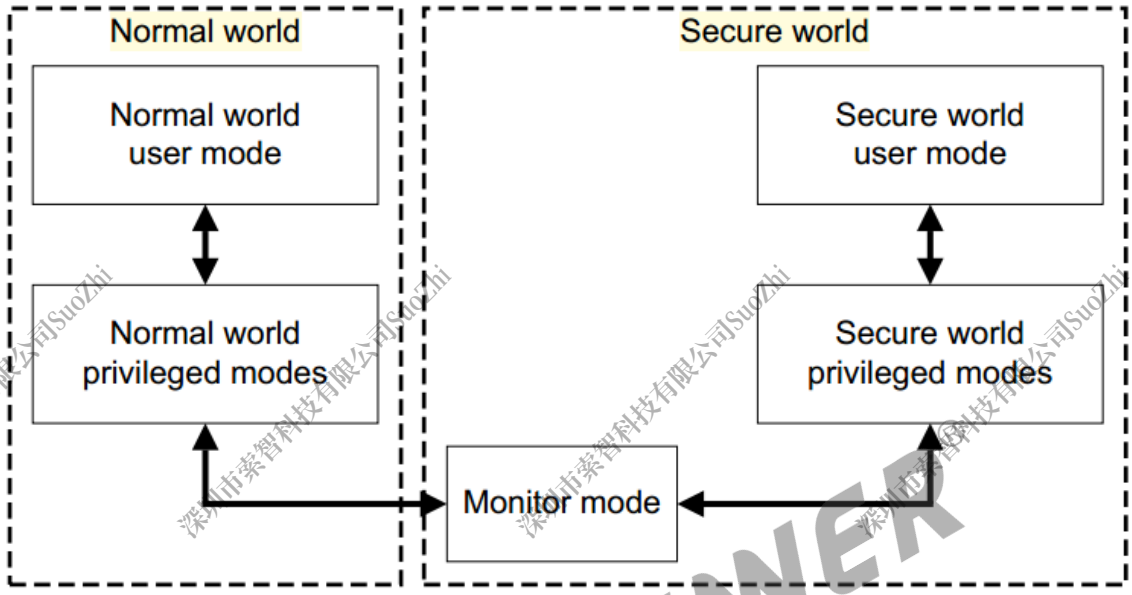

图 2-7: TurstZone 系统模型

TrustZone 技术将软硬件资源隔离成两个环境，分别为安全世界（Secure World）和非安全世界（Normal World），所有需要保密的操作在安全世界执行，其余操作在非安全世界执行，安全世界与非安全世界通过 monitor mode 来进行切换。具体可参考《TrustZone security whitepaper.pdf》。

2.3.1 OPTEE

很多公司基于 TrustZone 推出了自己的安全操作系统，各自有各自的实现方式，但是基本都会遵循 GP（GlobalPlatform）标准。GlobalPlatform 是一个跨行业的国际标准组织，致力于开发、制定并发布安全芯片的技术标准，以促进多应用产业环境的管理及其安全、可互操作的业务部署。

OPTEE 是 Linaro 联合其他几个公司一起合作开发的基于 ARM TrustZone 技术实现的 TEE 方案，遵循 GP 标准。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.4 相关术语

- SMC:Secure Monitor Call,ARM 给出的一条指令，可以让 CPU 从 Linux（非安全）直接跳转到 Monitor（安全）模式执行。
- RPC:Remote Procedure Control Protocol。OPTEE 中，用于操作 Linux 下资源的一种机制。比如,OPTEE 中不能读写文件，就通过 RPC 调用 Linux 下的文件系统来完成。
- REE:Rich Execution Environment。顾名思义，是资源丰富的执行环境，比如常见的 Linux,Android 系统等。
- TEE:Trusted Execution Environment。可信执行环境，即安全执行环境，在这个区域内，所有的代码，资源都是用户可以信任的。
- TA:Trusted Apps，在 TEE 下执行的应用程序，完成用户需要保护的任务，比如对密码的保护。
- CA:Client Apps，在 REE 下执行的应用程序，完成普通的，不需要保护的任务，比如看普通视频。
- UUID:Universally Unique Identifier，通用唯一识别码。由当前日期和时间，时钟序列，机器识别码（如 MAC）组成。

深圳市索智科技有限公司SuoZhi

版权所有 © 珠海全志科技股份有限公司。保留一切权利

3 硬件安全模块

TrustZone 技术要求安全非安全使用独立的外设资源。allwinner 平台上，安全非安全的资源的隔离通过指定的外设进行控制，具体有下面三个:

- spc(Secure Peripherals Control)

指定外设的安全属性，某外设被设定为安全后，该外设只有在安全世界才能正常访问，非安全世界写无效，读为 0。

- sid(Secure ID)

控制 efuse 的访问。efuse 的访问只能通过 sid 模块进行。sid 本身非安全，安全非安全均可访问。但通过 sid 访问 efuse 时，安全的 efuse 只有安全世界才可以访问，非安全世界访问的范围结果固定为 0。

安全的 efuse 存放安全的密钥如下:

密钥的名字	作用	使用场景
ROTPK	根证书的公钥的哈希值	用于 secure boot 的鉴权
SNVC	固件版本号	用于 secure boot 的固件防回滚
SSK	AES KEY，安全存储密钥	用于加密和解密 flash 的固件或第三方库文件
HUK	AES KEY，一机一密	用于加密和解密设备产生的敏感的数据

- smc(Secure Memory Control)

控制内存地址空间的访问。某地址空间的内存被设定为安全后，该空间内的内存只有安全世界可访问，非安全世界写无效，读为 0。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

4 secure boot

4.1 基本原理

secure boot 启动过程中, 芯片在启动时, 会先对系统做安全性检验, 检验通过后才引导系统。检查不通过则认为系统已经被修改, 拒绝引导系统并进入烧录模式。

安全性校验主要针对两个项目进行:

4.1.1 基于证书的哈希校验

固件中会包含证书, 证书记录了固件的哈希值。芯片验证证书有效后, 会使用证书记录的固件哈希值, 和实际计算得到的固件哈希值对比, 两者匹配则认为固件检验 OK。固件由多个子固件组成,brom->sboot->atf(64 位平台,32 位平台则跳过)->OPTEE->uboot->boot.img 的启动过程中, 每个子固件都有对应的证书用于该子固件的哈希校验。确保整个方案的安全启动。

固件中各子固件的具体验证流程如下:

1. 使用 efuse 中的 rotpk 哈希验证证书记录的 rotpk 的有效性。
2. 使用 rotpk 验证证书的有效性。
3. 使用证书中记录的公钥验证子固件对应证书的有效性。
4. 使用子固件对应证书记录的哈希值验证子固件的有效性。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

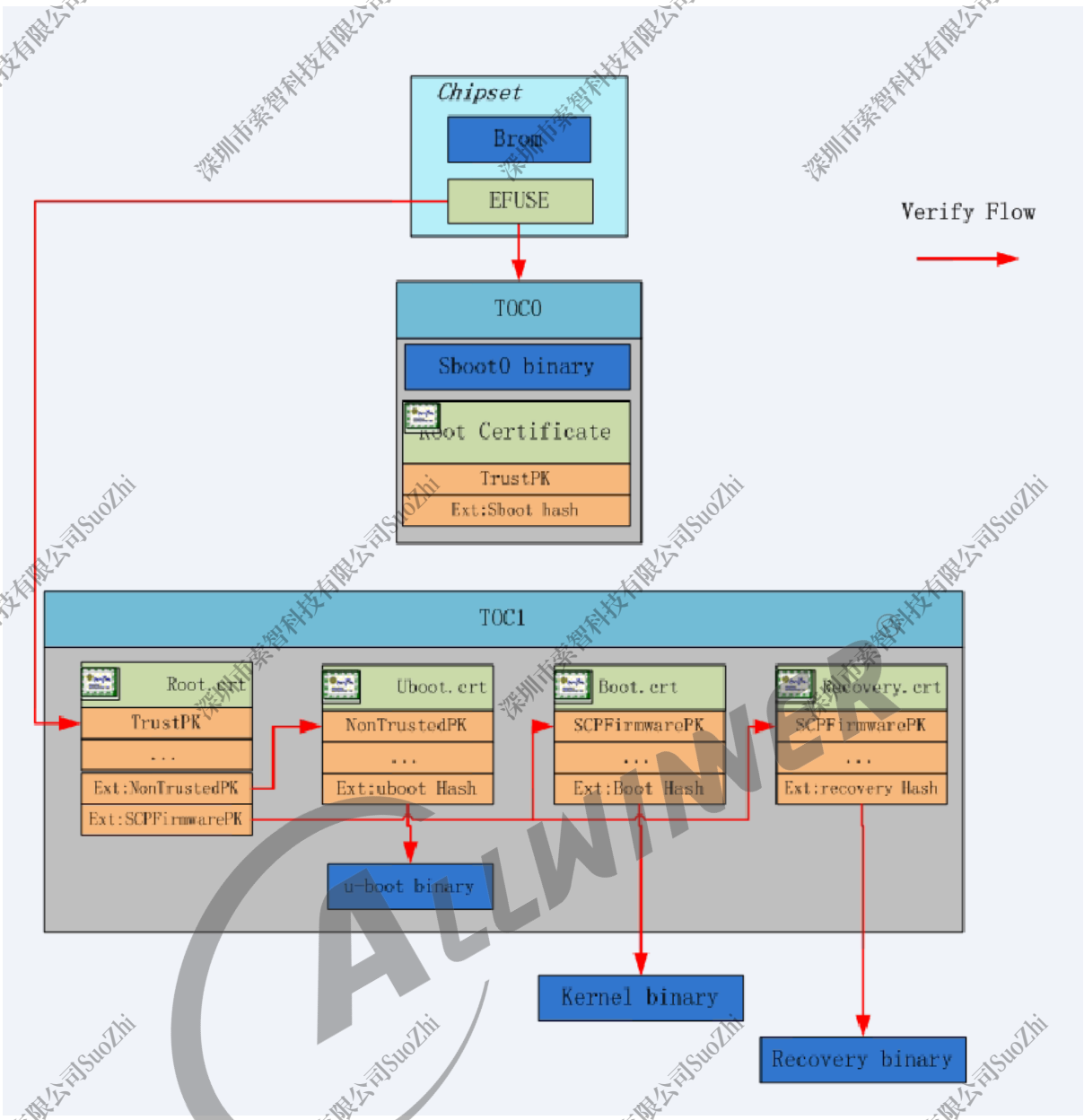

图 4-1: 固件验证过程

4.1.2 防回滚版本号校验

固件会含有一个用于防回滚检验的版本号，芯片会对比内部存储的版本号与固件的版本号。固件的版本号大于等于芯片记录的版本号时，认为固件的防回滚检验通过。

芯片会按需更新内部存储的版本号，确保存储的版本号为所有运行过的安全固件中，版本号最大的值。

下面介绍如何配置两个检验项目并生成安全固件，使能 secure boot。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

4.2 配置密钥

证书的生成与打包已经整合到安全固件的打包流程中，无需额外配置，只需要配置证书签名时使用的密钥对即可。配置方法有两种，第一种是使用提供的密钥对生成工具生成密钥；第二种是使用之前生成好的密钥。两种方法的具体说明如下：

a. 生成密钥

运行 build/createkeys 工具，选择对应平台后，即可自动生成密钥对。

生成的密钥位于 out/$(platform)/common/keys 目录下。

其中 rotpk.bin 为烧录到芯片中，用于验证根证书的公钥。Rotpk.bin 需要在烧录了安全固件的设备上才能烧录到芯片中，使用方法后述，详见 rotpk 烧写。其他为打包固件时用于为固件包签名的私钥。一个固件由多个部分组成，每个部分使用单独的密钥对进行签名认证。

⚠ 警告

这些密钥都是相互关联的，必须配套使用。生成的密钥请成套妥善保存

b. 使用已有密钥

如果之前已经生成过密钥，将密钥文件放到 longan/out/$(platform)/common/keys 目录下即可。

⚠ 警告

密钥的文件数量及名称都是根据平台固件打包的过程做过适配的。A 平台生成的密钥不可用于 B 平台安全固件的打包。否则打包过程可能会因为找不到指定的密钥而失败

4.3 配置防回滚版本号

芯片在引导固件的时候，会对比固件的版本号与芯片内存保留的版本号。

防回滚版本号在 longan/devices/configs/chips/${chip}/configs/default/version_base.mk 中进行配置，文件中主要有两个属性可配置：

版权所有 © 珠海全志科技股份有限公司。保留一切权利

```
c/d/version_base.mk
# define the verions of the image
# format: main.sub
# such as 1.01, 2.33
# NOTICE: the range of main version is from 0 to 31,
#         the range of sub  version is from 0 to 63
# when you change the version, you must increase one of them or bo
ions.
# the default version is 0.0

ROOT_ROLLBACK_USED = 1
MAIN_VERSION = 3
```

图 4-2: 防回滚版本号配置文件

- ROOT_ROLLBACK_USED

是否填充供 BROM 使用的返回滚版本号，平台相关，已经配置，使用默认值即可。

- MAIN_VERSION

固件的防回滚版本号，可用范围为 0-31。配置其他值芯片会直接认为固件版本号检验失败。

4.4 量产模式说明

量产模式又称为下载模式，用于烧写固件或者升级固件，量产模式的说明：

- 当 SBROM 启动失败时，会把启动模式切换位量产模式。
- 量产模式是非安全模式，即 CPU 和系统资源的属性都是非安全属性，因此无法访问安全的模块、安全的空间、以及安全的数据。
- 量产模式的永远为非安全模式，即无法切换到安全模式，因为 SBROM 不支持非安全切换安全的功能。

4.5 生成安全固件

在 longan 目录下运行./build.sh pack_secure 即可打包安全固件。

生成的固件在 longan/out/ 目录下，使用 phoneSuit 工具进行烧录。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

4.6 ROTPK 烧写

Rotpk 通过 PC 端工具 dragonSN 进行烧录。DragonSN 工具通过 usb 与设备通信，控制设备烧录指定的 rotpk 信息。具体烧录步骤如下:

1. 配置 burn_key 属性

设置 burn_key 属性值为 1 后，设备才会接收 DragonSN 通过 usb 传输的信息，进行相应的烧录工作。该属性在文件 devices/config/chips/*chips/configs/*{board}/sys_config.fex 中,[target] 项下，如图。如果未显式配置，按 burn_key=0 处理。

```
;----------------------------------------------------------------------------
[target]
boot_clock      = 1008
storage_type    = -1
burn_key = 1
;----------------------------------------------------------------------------
; power setting
```

图 4-3: 配置烧 key 属性

2. 打包安全固件并烧录，打包时使用的密钥必须与烧录的 rotpk 匹配，具体原因详见 rotpk 烧录时小机端的处理过程。
3. 在 PC 端配置 DragonSN 工具后运行。
4. 设备通过 usb 与 pc 连接后开机。
5. DragonSN 显示设备已连接后开始烧录（详见 DragonSN 工具的使用说明）。

- rotpk 烧录时小机端的处理过程。

为了保证不会误烧错误的 rotpk，烧录时小机端会做额外的处理，具体流程如下:pc 工具下发的 rotpk,uboot 会在确认与当前固件的 rotpk 匹配后才请求 Monitor 烧录该 rotpk。对于 64 为平台,Monitor 为 ATF，对于 32 位平台,Monitor 为 OPTEE。

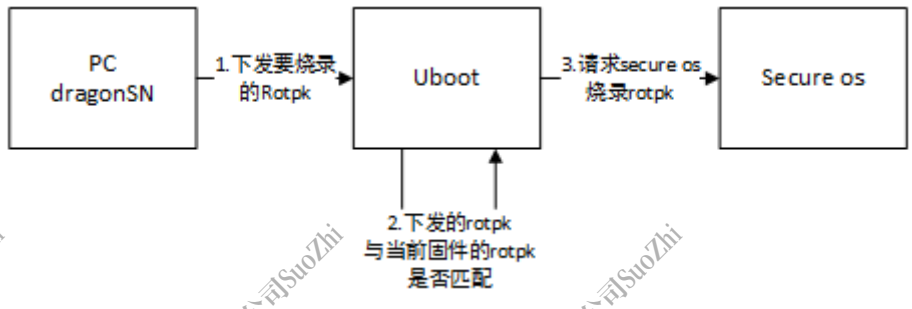

图 4-4: rotpk 烧录时小机端处理

版权所有 © 珠海全志科技股份有限公司。保留一切权利

4.7 注意事项

1. rotpk 只能烧录一次，烧录后不可再修改。请妥善保管《配置密钥-a. 生成密钥》中生成的密钥文件。
2. 烧录过安全固件后，芯片每次上电都会对固件进行安全性检查，这时候烧录普通固件，会因为无法通过检查而不能启动。故安全固件和普通固件不可混合使用。
3. 芯片未烧录 rotpk 时，跳过根证书上公钥的验证，继续进行后续的验证流程。
4. 芯片出厂时，芯片内记录的防回滚版本号为 0。

深圳市索智科技有限公司SuoZhi

ALLWINNER

版权所有 © 珠海全志科技股份有限公司。保留一切权利

5 secure os

5.1 OPTEE 介绍

Allwinner 软件平台采用 OPTEE 作为 secure os 的实现，它严格遵循 ARM Trust-Zone 和 TEE/GP 等产业标准。通过 OPTEE 来使用 TrustZone 技术。加入 OPTEE 后，运行时系统的总体构成如下图所示，下面介绍加入 OPTEE 后，系统中新增的关键构件。

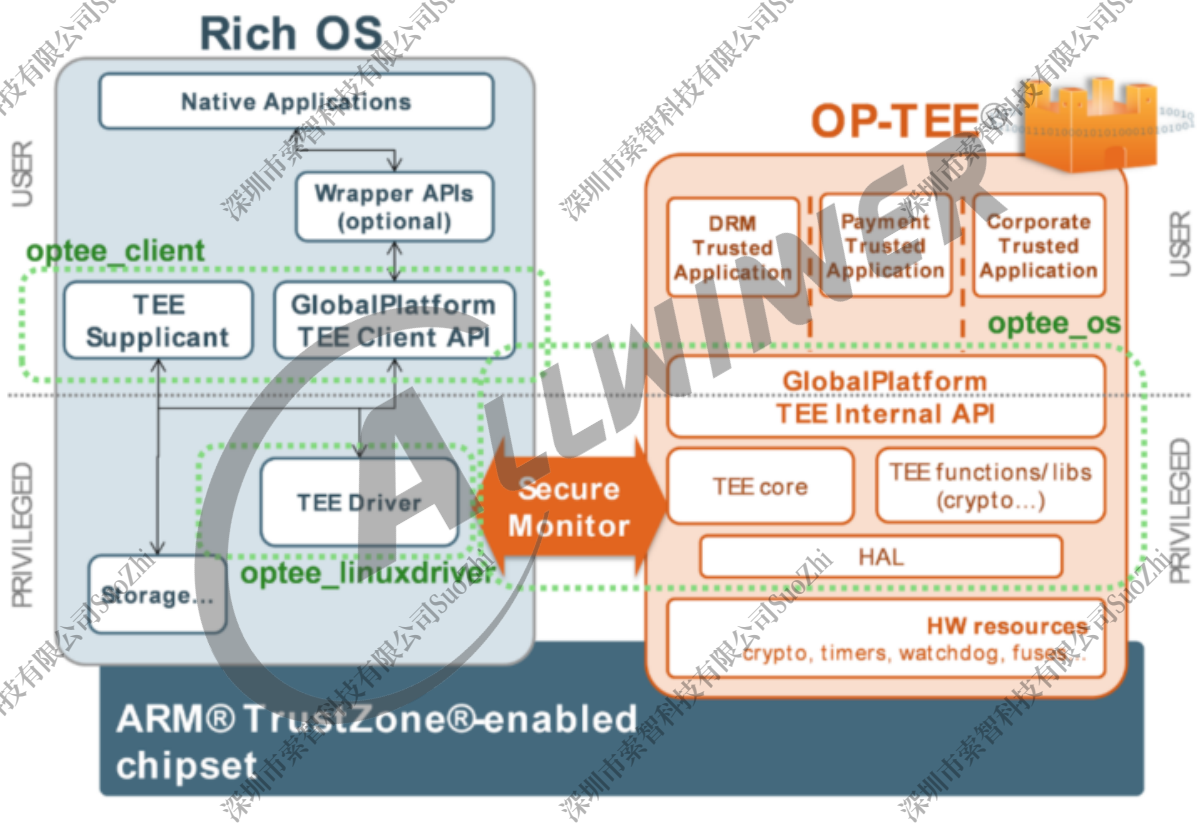

图 5-1: OPTEE 总体架构

5.1.1 OPTEE OS

为运行在 TEE 环境的 os。主要负责:

1. TEE 下的任务调度。
2. TEE 下的资源分配。
3. 为 TEE 下的应用程序提供系统调用，这些系统调用包含 GP 规定的 TEE internal API。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

5.1.2 OPTEE linux driver

REE 环境请求 TEE 环境的程序提供服务时，需要通过 TrustZone 指定的 smc 请求的形式进行。OPTEE linux driver 封装了这些请求。REE 对 TEE 环境的请求不直接操作硬件，通过对 OPTEE linux driver 的 IO ctrl 接口下发。

5.1.3 OPTEE client

OPTEE client 有两个用途:

a. 关于 REE 对 TEE 的请求,GP 有规定标准接口，称为 TEE client API。OPTEE client 实现这些接口,REE 环境下的应用程序只需要针对 TEE client API 接口进行编程即可完成对 TEE 的请求。
b. OPTEE OS 在运行时需要在请求 REE 环境协助进行文件操作。OPTEE client 接收这些请求并进行处理。如 TA（trusted apps，在 TEE 环境运行的应用程序）的加载。TA 存储在 REE 环境的文件系统，运行 TA 时,OPTEE OS 请求 REE 读取 TA 的数据后进行加载（已有针对 REE 下 TA 被窜改的预防措施）

5.1.4 TA/CA

TA:Trusted apps，运行在 TEE 环境的应用程序，敏感信息的直接处理在 TA 进行。

CA:Client apps，也称为 NA（Normal apps）。在 REE 环境下运行的应用程序，即传统的应用程序。

TrustZone 技术的使用基本以此形式实现:REE 下的应用程序请求 TEE 下的应用程序执行某些涉及敏感信息的操作:

如验证用户密码时,REE 提供用户输入的密码,TEE 比对用户输入的密码与设置的密码是否一致。返回比对通过/不通过。这样就在 REE 完全不接触设置的密码的明文的情况下实现了对输入的密码的验证。保证了已保存密码的安全。

这些功能都需要 TA/CA 配合实现,CA 作为 client 发起请求,TA 处理。

5.2 TEE 运行环境配置

为了 TA/CA 能够正常配合工作，满足产品需求。需要保证 TA/CA 运行时,TEE 的运行环境已经配置准备完毕。具体为 OPTEE OS、OPTEE linux driver、OPTEE client 已经配置妥当并正常运行。下面逐个组件说明如何配置。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

5.2.1 OPTEE OS

安全固件在编译时已经包含 OPTEE OS，且安全固件在进行 secure boot 的过程中会引导 OPTEE OS。无需额外配置。

5.2.2 OPTEE linux driver

a. 内核编译配置

内核编译时加入 linuxdriver，配置方式如下：

1. 在 longan 目录下输入./build.sh menuconfig 进入内核配置 UI。
2. 使能配置：

```
    General setup  --->
[*] Enable loadable module support  --->
-*- Enable the block layer  --->
    Platform selection  --->
    Bus support  --->
    Kernel Features  --->
    Boot options  --->
    Userspace binary formats  --->
    Power management options  --->
    CPU Power Management  --->
[*] Networking support  --->
    Device Drivers  --->
    Firmware Drivers  ----
    File systems  --->
[ ] Virtualization  ----
    Kernel hacking  --->
    Security options  --->
-*- Cryptographic API  --->
    Library routines  --->
```

图 5-2：进入驱动配置

版权所有 © 珠海全志科技股份有限公司。保留一切权利

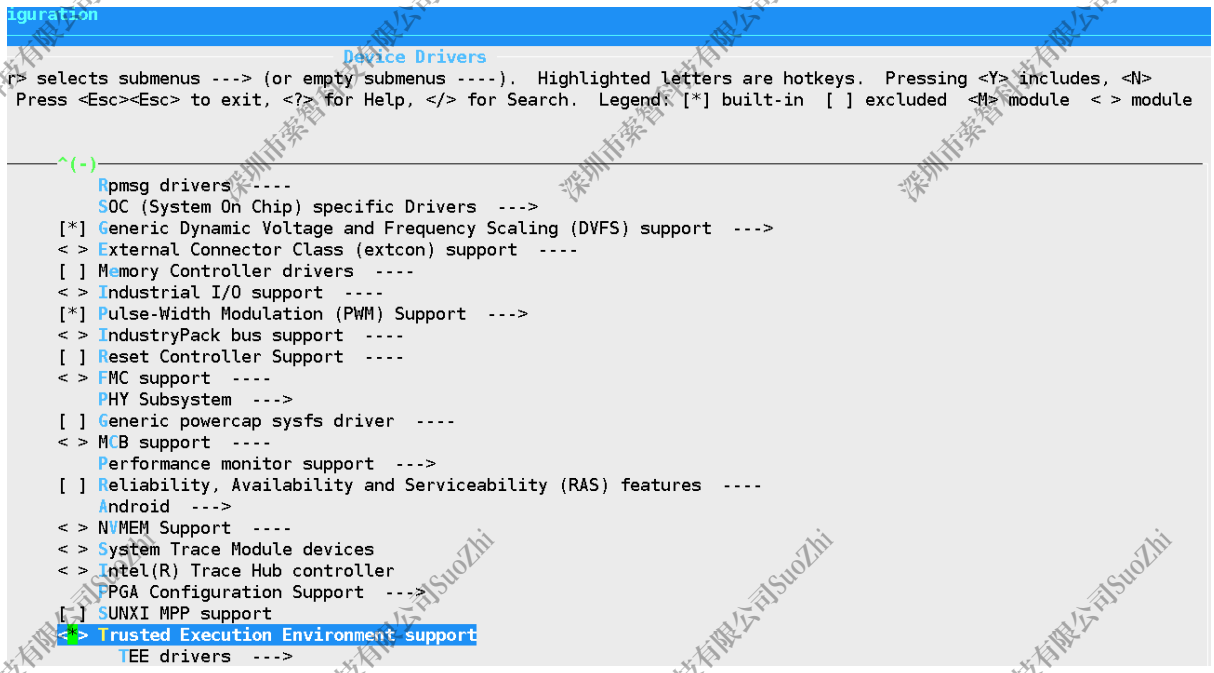

图 5-3: 使能 linux TEE driver

以及进入 TEE drivers —-> 勾选:

图 5-4: 使能 TEE driver 中 OP-TEE 的 driver

5.2.3 OPTEE client

a. 加入对应的动态库及 tee-supplicant 应用程序。

将表格中的文件拷贝到 longan/out/${platform}/linux/common/rootfs_def/对应文件夹下:

文件名	拷贝到
tee-supplicant	bin/
libteec.so libteec.so.1 libteec.so.1.0	lib/

版权所有 © 珠海全志科技股份有限公司。保留一切权利

警告

拷贝后，在 longan 目录下运行./build.sh 才会将相应的动态库及 supplicant 应用程序打包到根文件系统中。

b. 运行 tee-supplicant 应用程序。

1）方法一（手动启动）

在 linux 环境下, 切到 tee-supplicant 所在目录 (/bin), 执行:

```
./tee-supplicant &
```

警告

tee-supplicant 运行后会无限循环监听 TEE 的请求，不会返回。请不要漏掉 &。

2）方法二（自动启动）

修改 linux 初始化流程, 开机时自动启动。

5.2.4 TEE 环境内存空间配置

a. TEE 环境的内存使用。

TEE 环境使用的内存有 3 部分, 各部分大小在 optee os 编译时指定。各部分作用如下:

```
i. 共享内存:用于REE与TEE的数据交换,TrustZone技术中REE与TEE的交互通过smc进行,smc只能通过寄存器交换有限的
   数据,更多的数据通过共享内存进行交换。

ii. Optee os内存:optee_os专用的内存。optee_os加载到此处运行。

iii. TA内存堆:加载TA、放置TA堆、栈的内存空间。由optee_os进行分配。分配给某一个TA的内存只能由TA或
   optee_os访问,其他TA无法访问。
```

b. 内核为 TEE 预留内存。

共享内存内核和 TEE 都有访问权限, 这部分内容现在由 TEE 进行管理。如果内核也调配此内存, 会导致内存被误用,REE 与 TEE 交互异常。其他 os 管控的内存 REE 没有访问权限,REE 在申请到这些内存后会因为任何操作都只能读到 0 而运行异常。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

因此，需要配置内核让内核预留 TEE 环境使用的物理地址空间，避免内存访问的冲突。具体配置方法如下：修改文件 arch/arm64/boot/dts/sunxi/${platform}.dtsi，加入如下内容。SDK 中已经配置完毕，不需要修改。

linux-5.4:

```
reserved-memory {
    #address-cells = <2>;
    #size-cells = <2>;
    ranges;

    bl31 {
        reg = <0x0 0x48000000 0x0 0x01000000>;
    };
};
```

其他版本 linux:

```
/memreserve/ 0x48000000 0x01000000;
```

5.3 TEE 运行环境的使用

TEE 运行环境准备配置完毕后即可通过 TA/CA 使用基于 TrustZone 技术的功能。TA/CA 开发详见 TA/CA 开发指引。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

6 TA/CA 开发指引

6.1 开发环境目录结构

开发环境目位于 longan/tee_kit, 目录结构如下:

文件（夹）	说明
build.sh	编译脚本
dev_kit	编译依赖 (平台相关)
demo	demo
platform_config.mk	平台信息
tools	编译工具链

6.2 编译

a. 运行./build.sh -t, 解压编译工具链。

b. 运行./build.sh 编译所有 DEMO。

6.3 编译脚本使用说明

命令	功能
-h	显示帮助消息。
-t	解包 tools 目录中的编译工具链。
clean	清除所有 demo 编译输出。
config	选择编译平台。

6.4 拷贝

拷贝下列文件到设备:

版权所有 © 珠海全志科技股份有限公司。保留一切权利

- TA/CA 运行必须的公共文件

文件	拷贝到
./dev_kit/arm-plat-${platform}/export-ca/bin/tee-supplicant	不限定位置
./dev_kit/arm-plat-${platform}/export-ca/exportlib/libteec.so	/lib
./dev_kit/arm-plat-${platform}/export-ca/exportlib/libteec.so.1	/lib
./dev_kit/arm-plat-${platform}/export-ca/exportlib/libteec.so.1.0	/lib

- 实际的 TA/CA 程序

demo 对应的输出文件 (详见各 demo 的说明)

6.5 运行

6.5.1 运行辅助 REE 与 TEE 通信的守护进程

```
${Supp_dir}/tee-supplicant &
```

Supp_dir 为放置 tee-supplicant 的目录。

tee-supplicant 作为守护进程运行, 不会返回, 必须带"&"运行。

6.5.2 运行 DEMO

6.5.2.1 helloworld

本 demo 展示 CA 如何调用 TA, 以及如何通过共享内容向 TA 传输数据。

a. 拷贝的文件

文件	拷贝到
./demo/optee_helloworld/na/hello_world_na	不限定位置
./demo/optee_helloworld/ta/12345678-4321-8765-9b74f3fc357c7c61.ta	/lib/optee_armtz

版权所有 © 珠海全志科技股份有限公司。保留一切权利

b. 运行

命令	输出
hello_world_na	NA:init context NA:open session TA:creatyentry! TA:open session! NA:allocate memory NA:invoke command TA:rec cmd 0x210 TA:hello world! NA:finish with 0
hello_world_na 1234 注:{1234} 可以为任意字符串	NA:init context NA:open session TA:creatyentry! TA:open session! NA:allocate memory NA:invoke command: hello 1234 TA:rec cmd 0x210 TA:hello 1234 NA:finish with 0

6.5.2.2 encrypt_file_storage

本 demo 展示如何通过 TA 在 REE 的文件系统创建、读、写、删除加密文件。

a. 拷贝的文件

文件	拷贝到
demo/encrypt_file_storage/na/demo/demo demo/encrypt_file_storage/ta/2977f028-30d8-478b-975c-beeb3c134c34.ta	任意位置 /lib/optee_armtz

b. 运行

版权所有 © 珠海全志科技股份有限公司。保留一切权利

命令 文件名以 test 为例	输出
demo -c test	创建文件成功没有特殊输出
demo -w test 写的数据由 demo 随机生成	—- Write file:test with 256 Bytes data: —- 99 9c 9b 66 88 2c c8 c9 19 55 72 10 f7 c3 70 7f 1a 51 56 74 35 03 e4 6f 1b 40 4d 64 29 b5 ba c2 52 56 a8 db 03 f1 25 1c 47 97 ac bf da 1d 3f f4 ed 15 69 a3 18 cd 92 33 0f df 98 b7 15 d2 fa 67 a8 23 c2 ab 15 68 48 dc 00 f4 9c db 91 5b d0 80 70 ba a3 88 08 36 3b 96 16 53 ce aa 26 c9 12 4f ec 55 fa 82 bd c2 5f 3d b7 7b 98 4a 56 e9 4a c6 a4 ed ce 2c 24 89 c3 b9 dd 92 64 83 db f5 d2 48 ca 4e ca 08 11 a9 46 49 a4 de 93 fa c8 5d c1 ec 4b 10 1a ee 9a 5d 28 f7 6f 8c fa 4b 02 ce 13 cd 9c de d5 ad 08 9b 76 ad 7b 0a a8 c3 e6 ea 31 b1 7a 4b a0 94 28 c8 8c 97 d4 08 62 d7 56 75 25 f2 d3 7a 20 5c 17 97 0a 12 21 32 55 09 1d 86 ba 18 51 db ac 79 a4 b9 90 f9 c1 72 51 18 68 76 8a 3c 70 aa 98 07 c1 22 19 63 55 ee 6c f1 75 a6 89 c7 02 37 c0 27 f0 d1 21 32 c3 72 c9 2c 68 54 e8 d8 —- Write file:test end! —-

版权所有 © 珠海全志科技股份有限公司。保留一切权利

命令 文件名以 test 为例	输出
demo -r test	—- Read file:test 256 Bytes data: —- 99 9c 9b 66 88 2c c8 c9 19 55 72 10 f7 c3 70 7f 1a 51 56 74 35 03 e4 6f 1b 40 4d 64 29 b5 ba c2 52 56 a8 db 03 f1 25 1c 47 97 ac bf da 1d 3f f4 ed 15 69 a3 18 cd 92 33 0f df 98 b7 15 d2 fa 67 a8 23 c2 ab 15 68 48 dc 00 f4 9c db 91 5b d0 80 70 ba a3 88 08 36 3b 96 16 53 ce aa 26 c9 12 4f ec 55 fa 82 bd c2 5f 3d b7 7b 98 4a 56 e9 4a c6 a4 ed ce 2c 24 89 c3 b9 dd 92 64 83 db f5 d2 48 ca 4e ca 08 11 a9 46 49 a4 de 93 fa c8 5d c1 ec 4b 10 1a ee 9a 5d 28 f7 6f 8c fa 4b 02 ce 13 cd 9c de d5 ad 08 9b 76 ad 7b 0a a8 c3 e6 ea 31 b1 7a 4b a0 94 28 c8 8c 97 d4 08 62 d7 56 75 25 f2 d3 7a 20 5c 17 97 0a 12 21 32 55 09 1d 86 ba 18 51 db ac 79 a4 b9 90 f9 c1 72 51 18 68 76 8a 3c 70 aa 98 07 c1 22 19 63 55 ee 6c f1 75 a6 89 c7 02 37 c0 27 f0 d1 21 32 c3 72 c9 2c 68 54 e8 d8 —- Read file:test end! —-
demo -d test	Delete file:test !
demo -h	显示帮助信息

深圳市索智科技有限公司SuoZhi

6.5.2.3 base64-usage

本 demo 展示提供的 base64 软实现如何使用。

a. 拷贝的文件

文件	拷贝到
./demo/base64-usage/na/base64_demo ./demo/base64-usage/ta/b0e8fef8-b857-4dd4-bfa6088373069255.ta	不限定位置 /lib/optee_armtz

b. 运行

命令	输出
base64_demo -e 123456 -e 后为需要编码的字节串, 两个字符对应一个字节	input bytes: 0x12 0x34 0x56 NA:open session TA:creatyentry! TA:open session! NA:allocate memory TA:rec cmd 0x221 input size:3 encode result: EjRW NA:finish with 0 base64_demo -d EjRW -d 后为需要解码的字符串

- 提供给 TA 的 base64 实现的接口

接口	说明

size_t EVP_EncodeBlock(uint8_t *dst, const uint8_t* src,size_t src_len);
 	
int EVP_EncodedLength(size_t *out_len, size_t len);
int EVP_DecodedLength(size_t out_len, size_t len);*
	对 src 开始长为 src_len 的 u8 数组进行编码, 输出到 dst, 输出的字符串用'\0' 结尾。返回值为编码后字符串长度, 不含结尾的'\0'
	计算 len 长度的输入在编码/解码后的输出长度, 写到 out_len 中。计算成功返回 1, 计算失败返回 0

版权所有 © 珠海全志科技股份有限公司。保留一切权利

接口	说明
int EVP_DecodeBase64(uint8_t *out, size_t* out_len,size_t max_out, const uint8_t *in,size_t in_len);	对 in 开始长为 in_len 的输入进行解码，输出到 out，解码后的长度填入 out_len,max_out 为 out 对应的 buffer 的大小。成功返回 1，失败返回 0。

6.6 编译配置

下面介绍开发新的 TA/CA 时如何配置依赖，篇幅有限，这里只提一些关键点，具体请参考 demo 的源码。

6.6.1 TA

a. 在 TA 源码的根目录创建 Makefile，包含以下内容：

内容	说明
BINARY=$(UUID)	最终生成的.ta 文件的文件名,TEE 加载 TA 时已 UUID 为文件名读取文件，必须与 TA 的 UUID 一致
include ${DIR}/ta_dev_kit.mk	TA 代码编译的二进制有特殊处理的需要，使用 dev_ki/arm-plat-${chip}/export-ta_arm32/mk/ta_dev_kit.mk 进行编译

b. 在源码的根目录创建 sub.mk，提供编译信息：

命令	作用
srcs-y += filename	filename 加入编译
subdirs-y += dirname	include dirname 下的 sub.mk
global-incdirs-y += dirname	编译 TA 时头文件搜索目录
cflags-/-y +=	c 文件编译 flagfilename 为生效的文件,cflags-y 对所有 c 文件有效
aflags-/-y +=	s 文件编译 flag
cppflags-/-y +=	c 及 s 文件编译 flag
TA_PRIVATE_FLAGS +=	连接生成 TA 的 elf 时使用的 flag

c. 创建 user_ta_header_defines.h 文件提供 TA 的信息

版权所有 © 珠海全志科技股份有限公司。保留一切权利

OPTEE OS 在接在 TA 时会根据这里的配置项给 TA 配置运行环境。

如 TA 的堆栈,TA 可用的堆栈大小在此时固定, 加载后不可再调整。如果 TA 需要使用较大的堆栈空间, 请调整 TA_STACK_SIZE、TA_DATA_SIZE 的值, 让 OPTEE OS 为 TA 分区更多堆栈空间。

宏	作用
TA_UUID	TA 的 uuid, 与 $(BINARY) 一致
TA_FLAGS	运行配置, 使用 demo 默认值即可。 完整选项见 dev_kit/arm-plat-${chip}/export-ta_arm32/include/user_ta_header.h
TA_STACK_SIZE	栈大小
TA_DATA_SIZE	堆大小

d. 包含描述 TEE TA API 的头文件并提供 TEE TA API 要求的实现

实现	描述
TEE_Result TA_CreateEntryPoint(void)	打开 TA 的回调
void TA_DestroyEntryPoint(void)	关闭 TA 的回调
TEE_Result TA_OpenSessionEntryPoint(uint32_t nParamTypes, TEE_Param pParams[4], void **ppSessionContext)	创建新 session 时的回调
void TA_CloseSessionEntryPoint(void *pSessionContext)	关闭 session 时的回调
TEE_Result TA_InvokeCommandEntryPoint(void *pSessionContext, uint32_t nCommandID, uint32_t nParamTypes, TEE_Param pParams[4])	执行命令的回调

6.6.2 CA

a. 包含描述 TEE client API 的头文件。

b. 调用 TEE client API 与 TA 交互, 详见 demo。

c. 将 dev_kit/arm-plat-${platform}/export-ca/include 加入包含目录。

d. 将 dev_kit/arm-plat-${platform}/export-ca/exportlib 加入库搜索目录。

e. 使用 `"-lteec"`选项连接实现 TEE client API 的动态库 libteec.so。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

6.6.3 TA 加密

a. 打开/关闭 TA 加密

对于支持 TA 加密的平台, 在使用./build.sh config 选中该平台后会提示 encrypt TA(y/n): ,y、n 对应是否对 TA 进行加密,secure os 加载 TA 时会自动判断 TA 是否需要解密。

b. 配置加密密钥

TA 加密使用通过 aes 进行, 密钥长度为 128bit(16 字节), 存放在 dev_kit/arm-plat-${platform}/export-ta_arm32/keys/ta_aes_key.bin 文件中。如果需要修改 TA 加密使用的密钥, 请修改此 bin 文件。

芯片使用 efuse 中的 ssk key 对 TA 进行解密。请确保此 aes key 与芯片的 efuse 中的 ssk key 一致。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

7 密钥存储

无论是使用密钥进行加密解密，还是使用哈希校验固件。都涉及到对密钥、对哈希的保存。这些信息以 key 的形式烧录到设备中。烧录过程通过 dragonSN 工具进行。key 保存的方式有多个，每种保存方式的特性各不相同。下面根据 key 存储的方式进行介绍。

7.1 efuse

保存在芯片内的 efuse 上，不可擦除。通过 sid 模块访问。efuse 中的内容分成两部份，安全和非安全。访问 efuse 的读取结果视芯片状态而定，具体如下：

芯片状态	访问非安全 efuse	访问安全 efuse
安全	正常	正常
非安全	正常	返回 0

- 烧录配置

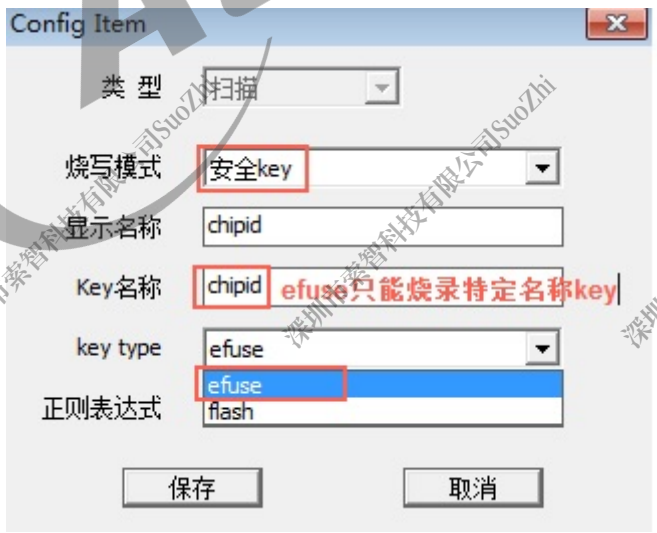

图 7-1: DragonSN 烧录安全 efuse 配置

- efuse map

efuse 空间在芯片设计时划分，只能烧录特定名称的 key。每个 key 在每个芯片 efuse 的存储位置请查阅芯片的 efuse map

版权所有 © 珠海全志科技股份有限公司。保留一切权利

7.2 flash

保存芯片外的 flash 上, 根据保存的 flash 区域的访问方式, 分为安全 key 和私有 key 两种

7.2.1 安全 key(secure storage)

保存在 flash 上, 使用的扇区未映射到逻辑扇区。通常的 flash 操作无法访问。正常量产不会被擦除。

- 烧录配置

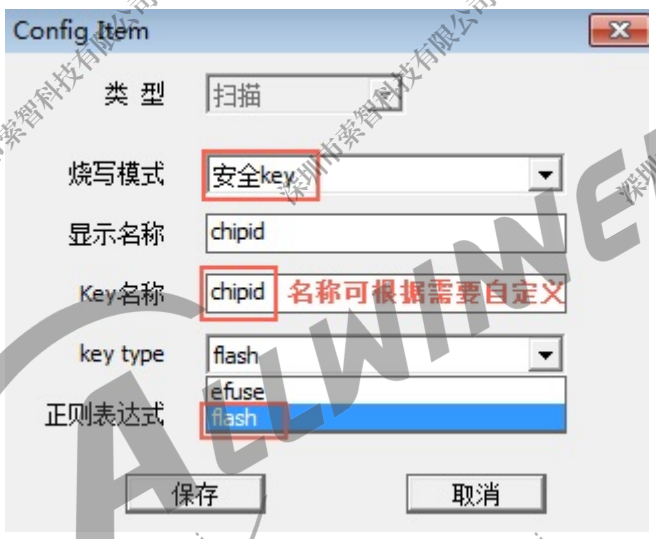

图 7-2: DragonSN 烧录安全 key 配置

7.2.1.1 keybox

特殊的安全 key, 这里保存的 key 不会保存明文, 而是保存由 secure os 加密过的密文。请注意,OPTEE OS 使用 efuse 中的 SSK key 来加密明文数据, 为了保证数据安全, 请在烧录 SSK 后进行 keybox 数据的烧录、使用。

a. keybox 列表

keybox 位于 secure storage 中。dragonSN 烧录工具只能指定 key 烧录到 secure storage。key 是否烧录到 keybox 中, 由 uboot 根据 key 的名称决定。

uboot 通过环境变量 keybox_list 判断当前烧录的 key 是否为需要保存到 keybox 的 key, 是则在烧录、加载时做相应的处理。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

keybox_list 环境变量位于 devides/config/chips/${chip}/configs/default/env.cfg 中，使用逗号分各 key. 下面的例子中，烧录时名称为 rsa_key 或 ecc_key 或 testkey 的 key 会保存到 keybox 供 secure os 解密使用。

```
keybox_list=rsa_key, ecc_key, testkey
```

b. 烧录

通过 dragonSN 工具进行烧录

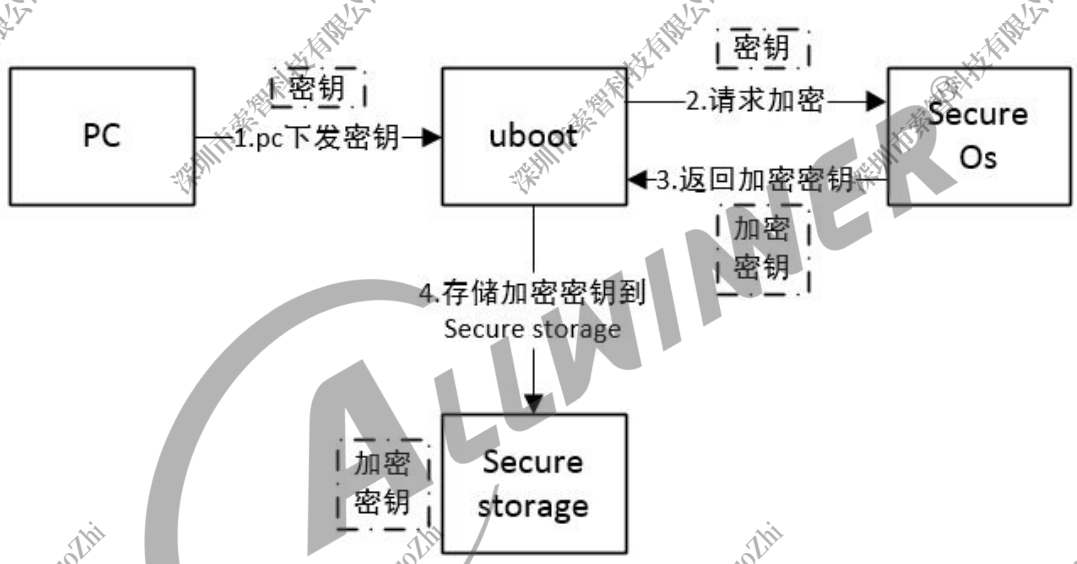

图 7-3: key 烧录到 keybox 的过程

c. 上电加载

uboot 读取 key 后送到 secure os。secure os 解密后缓存，供 TA 通过接口访问。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

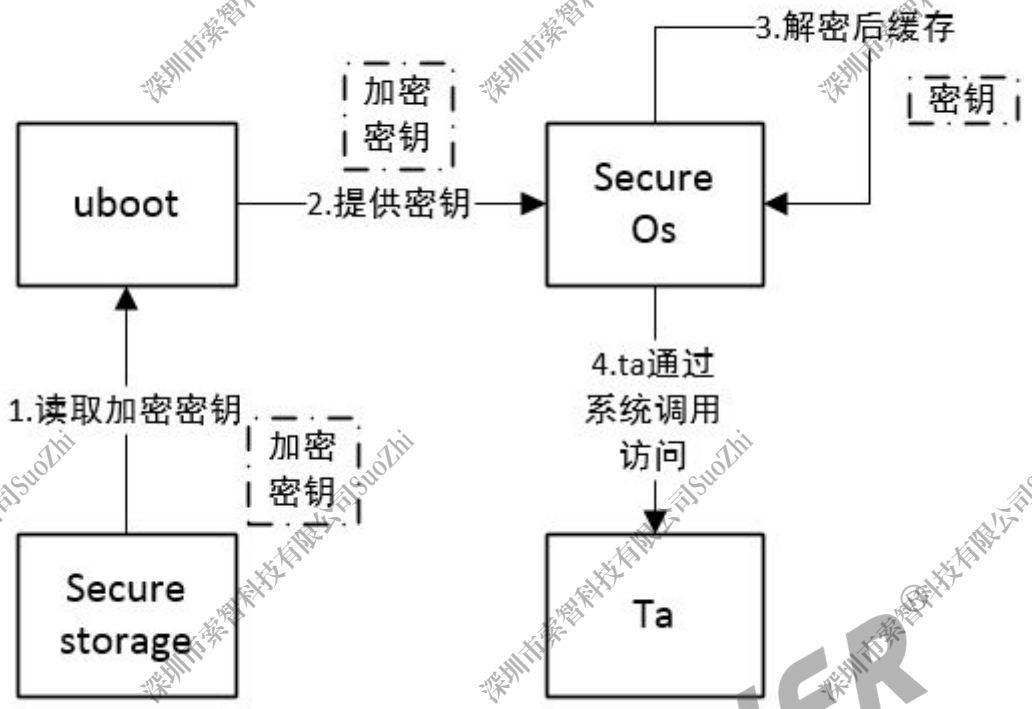

图 7-4: key 从 keybox 中加载的过程

7.2.2 私有 key(private storage)

保存在 private 分区。量产时会把此分区内容读取到内存, 量产完毕后从内存恢复到 flash 中, 达到量产后分区内容保留的目的。

key 保存到文件名为 key 名称的文件中, 可以通过文件系统访问到 key 的内容。

- 烧录配置

深圳市索智科技有限公司SuoZhi

图 7-5: DragonSn 烧录私有 key 配置

- 分区格式化

量产后 private 分区内容保持不变。故 private 分区无法通过量产进行格式化。对于未格式化的分区，烧 key 时需要进行分区格式化。

深圳市索智科技有限公司SuoZhi

图 7-6: 格式化私有分区

- 文件路径

key 保存到文件名为 key 名称的文件中, 文件路径可以在烧录时配置。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

图 7-7: 私有 key 烧录路径

:::warning 部分平台可能会读取特定私有 key, 读取时使用的路径为烧录私有 key 的默认路径。如果修改过 key 的烧录路径, 请注意同步修改涉及到的配置项。:::

在 sys_config.fex 中使用 $(keyname)_filename= `"$(path_to_key_file)"` 的形式指定 key 的读取。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

8 TEE 环境中数据的掉电保存

TrustZone 要求安全与非安全资源在硬件上进行隔离，其安全资源集成到芯片内以应对替换 pcb 上非芯片元件的攻击。但由于成本原因，大部分设备都没有集成到芯片内部的、大容量的掉电不丢失的、可擦写存储介质（如 nand），因此 TEE 中需要掉电保存的数据都需要保存到芯片外的存储介质中。对此 TrustZone 技术规范也有相应的方案,TEE 的中的数据需要保存到芯片外存储介质上时，需要满足指定要求。optee 提供了满足该要求的实现，可以用于 TEE 环境中数据的掉电保存。

8.1 OPTEE Secure Storage 功能框架

OPTEE Secure Storage 的软件架构如下:

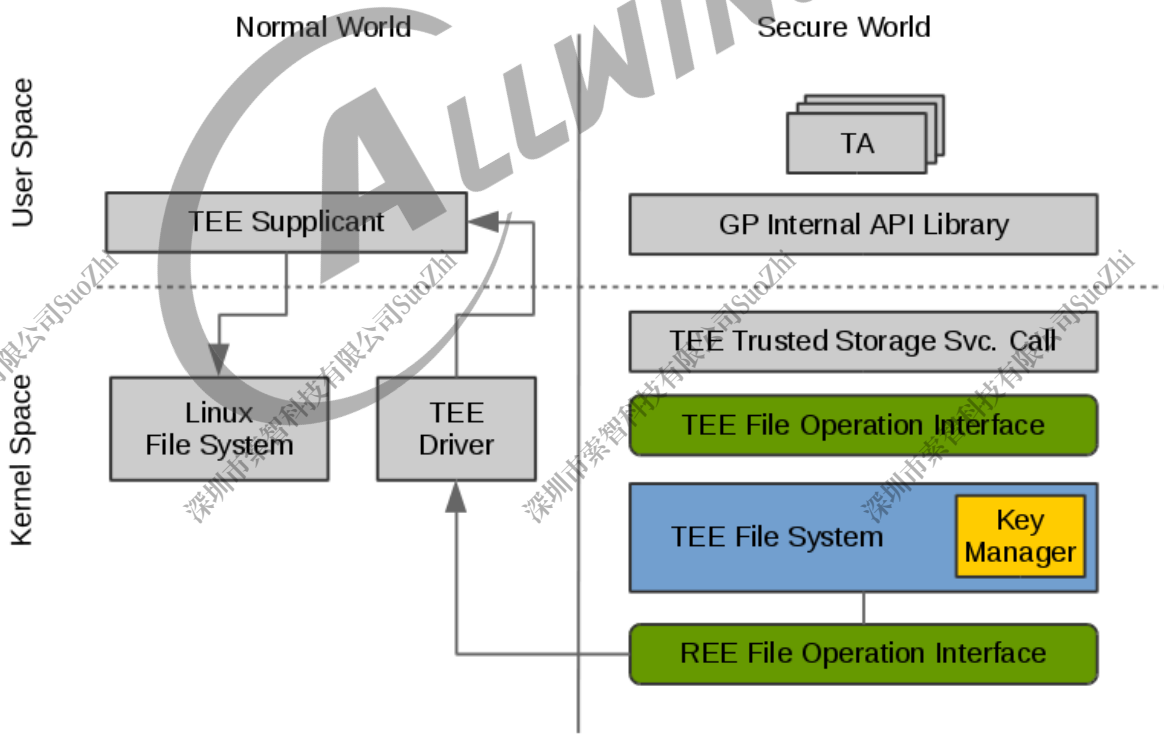

图 8-1: OPTEE Secure Storage 软件架构

版权所有 © 珠海全志科技股份有限公司。保留一切权利

8.2 文件操作流程

TA 调用 GP Trusted Storage API 提供的写接口，此接口会调用 TEE Trusted Storage Service 中的相关 syscall 实现陷入到 OPTEE 的 kernel space 中，该 syscall 会调用一系列的 TEE File Operation Interface 接口来存储写入的数据。TEE 文件系统会将写入的数据进行加密，然后通过一系列的 RPC 消息向 TEE supplicant 发送 REE 文件操作命令以及已加密的数据。TEE supplicant 对这些消息进行解析，按照参数的定义将加密的数据存放到对应的 Linux 文件系统中（默认是/data/tee 目录）。以上是对写数据的处理，对读数据的处理类似。

8.3 使用 OPTEE Secure Storage 为 REE 提供加密文件存储

配置 TA 作为服务端调用 OPTEE OS 提供的存储接口，可以让 REE 也使用此接口，完成数据的加密存储。具体可参考 demo 中的 encrypt_file_storage

深圳市索智科技有限公司SuoZhi

版权所有 © 珠海全志科技股份有限公司。保留一切权利

9 参考资料

a. TrustZone

1. PRD29-GENC-009492C_trustzone_security_whitepaper.pdf

b. GlobalPlatform

1. GPD_TEE_SystemArch_v1.1.pdf
2. GPD_TEE_Client_API_v1.0_EP_v1.0.pdf
3. GPD_TEE_Internal_Core_API_Specification_v1.1.pdf
4. GPD_TEE_TA_Debug_Spec_v1.0.pdf

c. OPTEE

1. https://www.op-tee.org/documentation/
2. optee_os/documentation/secure_storage.md

深圳市索智科技有限公司SuoZhi

版权所有 © 珠海全志科技股份有限公司。保留一切权利

著作权声明

版权所有 ©2022 珠海全志科技股份有限公司。保留一切权利。

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

ALLWINNER、ALLWINNER 全志科技、全志科技、ALLWINNER（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。

深圳市索智科技有限公司SuoZhi