Secure boot相关配置。

1, 概述

对于secure boot项目来说，配置secure boot策略其实就是根据mtk对应文档来配置的。

比如：Secure\_2.0\_Configuration\_SOP.pdf 和

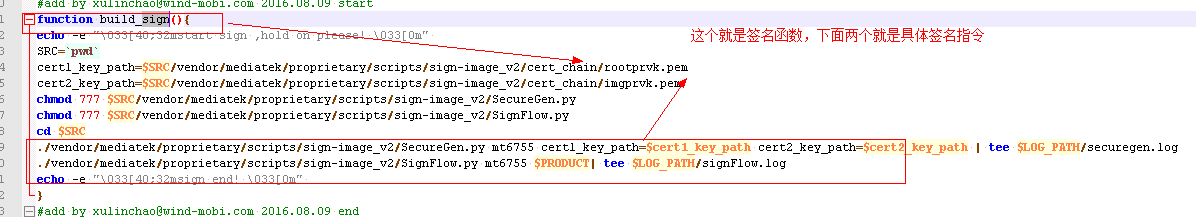
Security\_2.0\_Customization\_Guide\_-\_HW\_Root\_of\_Trust.pdf这类文档。

2, 配置方法

在配置secure boot的时候，先明确自己采用的secure boot哪一种，至少现在37平台采用的secure boot 1.1 。而50平台采用的事secure boot 2.0。secure boot具体配置方法，其实就是在本地生产一个1024的RSA和2048的RSA密钥来替换mtk原有的。如何操作，请参考对应mtk文档，在这里不做累述。

3,脚本的配置

在这里主要是修改三个脚本，分别是Auto,quick\_build.sh,release。这个作用主要是对镜像进行签名。最主要的就是quick脚本，在编译后，对生成的镜像进行签名。比如6750的平台，签名如下：

****

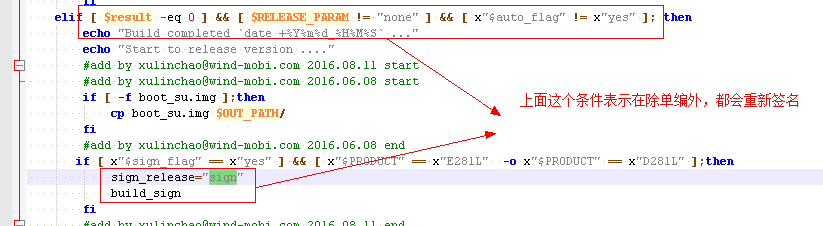
Auto脚本主要是加个标志位，能够自动编译签名版本。release脚本就是能将对应的签名文件释放出来。其具体代码，请参考286或者280等对应脚本的写法。286是针对secure boot 2.0做法，280 or 281是针对secure boot 1.1的做法。对应代码有些许差异。

4.注意事项

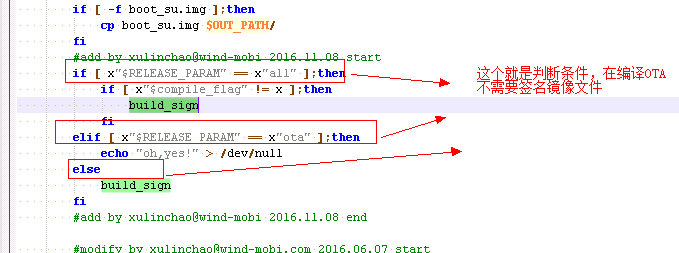
其实secure boot项目，最主要的就是这点。上面那些也没啥好说的，按照文档来做。

第一点：

在编译fota包OTA包之后，release出来的镜像刷机失败。这个是针对secure boot 1.1,y因为编译OTA之后，系统会重新打包镜像，所以我们必须重新签名镜像文件，

** 【secure boot 1.1】**

对于secure boot 2.0来说，编译OTA后，不需要重新签名镜像，因为python脚本已经做了处理。

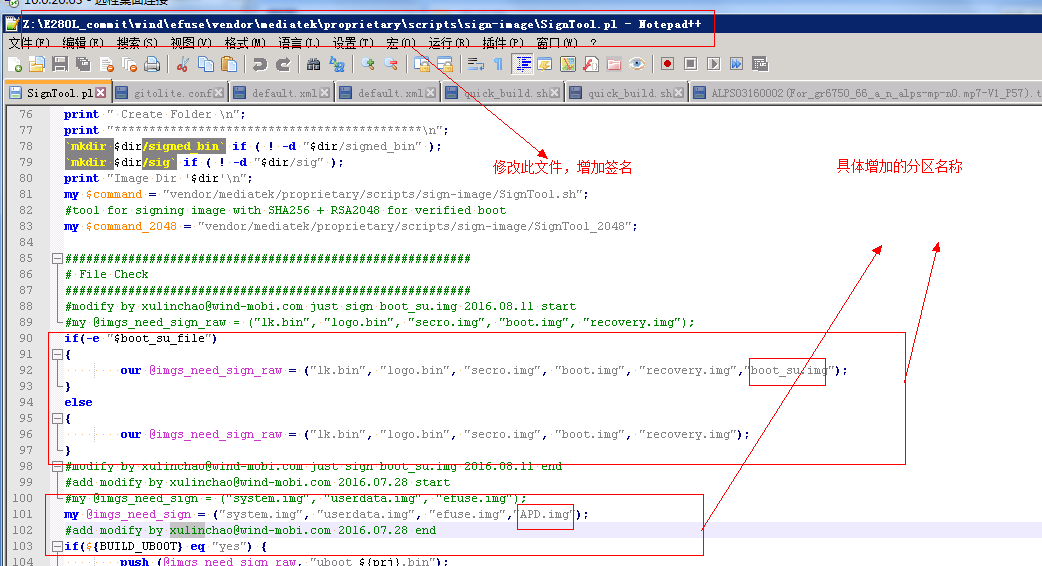
****

**【secure boot 2.0】**

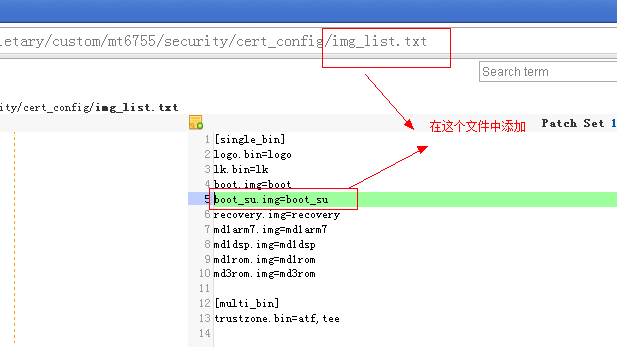
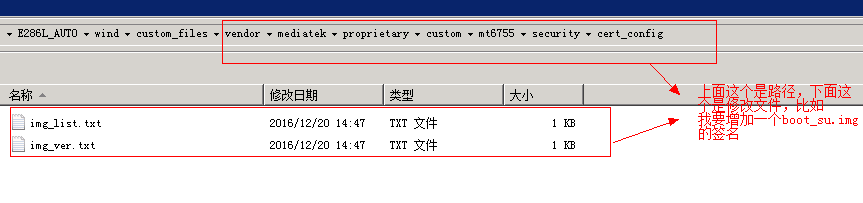
第二点：

就是我们如何增加新的签名文件？

对于secure boot 1.1来说

****

**对于secure boot 2.0来说**

****

****

对于secure boot 2.0来说，增加的签名文件不能超过32M，不然为编译报错。

刷机的时候，如果有新增加的分区，且超过了32M，不能签名镜像，那必须在flash 工具中跳过校验。

如何跳过，参考Security 2.0 Customization Guide - HW Root of Trust.pptx