#命令行签名工具用户手册

发布版本：1.0

作者邮箱：liuyi@rock-chips.com

日期：2021-12-27

文档密级：公开资料

前言3399|3328

概述

命令行签名工具为开发人员提供了固件|Loader|Uboot|Trust|Boot|DDR测试文件的签名和校验功能。

支持芯片
3588|3566|3568|3308|3326|3399|3229|3228h|3368|3228|3288|px30|3328|1808|3228P|1109|1126|2206

读者对象

本文档主要适用于开发人员

修订记录

日期	版本	工具版本	作者	修改说明
2021-12-27	V1.0	V1.0	刘翊	初稿

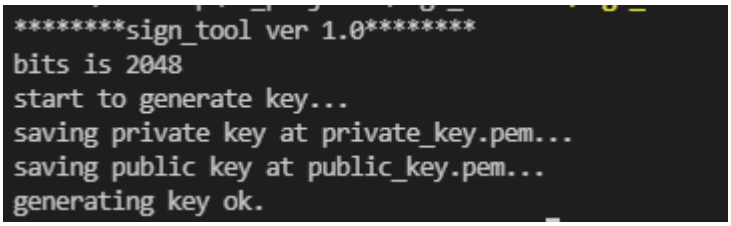
使用算法

芯片	摘要算法	签名算法	签名填
3229|3228h|3368|3228|3288|3399|3328	SHA256	RSA2048	NO PADD
3588|3566|3568|3308|3326|px30|1808|3228P|1109|1126|2206	SHA256	RSA2048	PSS PADD

1. 签名前准备

1.1 生成RSA公私钥对

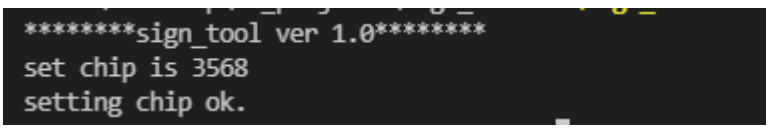
```
//产生rsa2048公私钥
sign_tool kk --bits=2048
or
sign_tool kk
```



1.2 选择芯片类型

签名前需要先选定芯片，选择会被记录，后续签名操作的芯片没有变化可以不用再选择

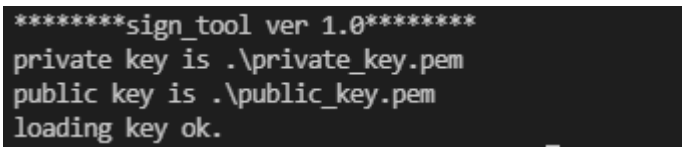
```
//例:选择1126芯片
sign_tool cc --chip=1126
```



1.3 选择RSA公私钥

签名前需要先选定使用的RSA密钥,选择会被记录，后续签名操作的密钥没有变化可以不用再选择

```
//加载私钥private_key.pem，公钥public_key.pem
sign_tool.exe lk --key private_key.pem --pubkey public_key.pem
```



1.4 签名Flag配置

可选功能，主要是进行Secureboot功能相关参数的设置

```
//例:开启miniloader写OTP功能
sign_tool ss --flag=0x20
```

```
********sign_tool ver 1.0********
sign flag is 0x20
setting sign argument ok.
```

1.5 设置中间数据保存路径

这个参数是一个目录路径，签名过程会将中间路径保存在此目录，或者是从此目录读取输入数据。主要使用在外部签名的情况

```
sign_tool ss --out .\data_out
```

2.签名操作

2.1 Update固件签名

Update固件的签名过程会对固件包中的loader,uboot,boot,trust等内容进行签名，由于过程涉及到update固件的解包和重新打包，时间会比较长。

****注:如果方案使用了ab分区，那么boot镜像的签名不通过工具来执行，所以请打开工具下的setting.ini文件，将exclude_boot_sign=true**

```
//签名update.img
sign_tool sf --firmware update.img
//校验已签名的update.img
sign_tool vf --firmware update.img
```

2.2 Loader签名

Loader的签名过程会对Loader文件进行解包插入生成的安全结构再重新打包。

```
//签名Loader
sign_tool sl --loader miniloader.bin
//校验已签名的loader
sign_tool vl --loader miniloader.bin
```

2.3 IDBlock签名

IDBlock是系统启动块包括引导数据结构，DDR初始化代码和Boot代码，由mkimage工具生成

```
//签名IDBlock
sign_tool sb --idb 1126_idblock.bin
//校验已签名的IDBlock
sign_tool vb --idb 1126_idblock.bin
```

2.4 DDR测试文件签名

```
//签名DDR测试文件
sign_tool sd --cfg 1126_ddr_test.cfg
//校验已签名的DDR测试文件
sign_tool vd --cfg 1126_ddr_test.cfg
```

2.5 镜像签名(uboot|trust|boot)

```
//签名uboot.img
sign_tool si --img uboot.img
//校验已签名的uboot.img
sign_tool vi --img uboot.img
```

3. 外部签名操作

3.1提取签名数据

此操作可以只指定公钥，提取出所有需要签名的数据到指定目录

```
//设置签名提取标志
sign_tool ss --extract
//设置待签名数据的提取存放目录
sign_tool ss --out wait_sign_dir
//提取update.img中的签名数据(可以是所有章节2中的签名操作)
sign_tool sf --firmware update.img
```

3.2 进行外部签名

对wait_sign_dir目录中的所有文件进行私钥签名，签名后的数据保存成同名文件放置wait_sign_dir下

```
//openssl进行pss填充和rsa 2048签名的方法
openssl pkeyutl -sign
-in to_sign.bin -inkey privatekey.pem
-out signed.bin
-pkeyopt digest:sha256
-pkeyopt rsa_padding_mode:pss
-pkeyopt rsa_pss_saltlen:-1
//openssl进行不填充和rsa 2048签名的方法
1.创建256字节buffer,填充0
2.拷贝sha256摘要到尾部32字节
3.使用openssl进行rsa2048签名
openssl rsautl -sign -in to_sign.bin -inkey privatekey.pem -raw -out signed.bin
```

3.3 注入签名数据

此操作会将wait_sign_dir中的所有签名数据注入到签名位置

```
//设置签名注入标志
sign_tool ss --inject
//注入签名数据到update.img中
sign_tool sf --firmware update.img
```