

安全工具用户手册

[版权声明](#)

[修订历史](#)

[目录](#)

[1. 工具总览](#)

[2. 工具介绍](#)

[2.1 product](#)

[2.2 efuse utility](#)

[2.3 fastboot](#)

[2.4 sectool](#)

[2.4.1 签名证书](#)

[2.4.2 imagesign](#)

[2.4.2.1 生成UBOOT签名镜像](#)

[2.4.2.2 生成TF签名镜像](#)

[2.4.2.3 生成TEE签名镜像](#)

[2.4.3 bin2ext4](#)

[2.4.3.1 生成uboot.EXT4](#)

[2.4.3.2 生成TF.EXT4](#)

[2.4.3.3 生成TEE.EXT4](#)

[2.4.4 KDF_GEN](#)

[3. 用户手册](#)

版权声明

Copyright © 2022 T-HEAD Semiconductor Co.,Ltd. All rights reserved.

This document is the property of T-HEAD Semiconductor Co.,Ltd. This document may only be distributed to: (i) a T-HEAD party having a legitimate business need for the information contained herein, or (ii) a non-T-HEAD party having a legitimate business need for the information contained herein. No license, expressed or implied, under any patent, copyright or trade secret right is granted or implied by the conveyance of this document. No part of this document may be

reproduced, transmitted, transcribed, stored in a retrieval system, translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise without the prior written permission of T-HEAD Semiconductor Co.,Ltd.

Trademarks and Permissions

The T-HEAD Logo and all other trademarks indicated as such herein are trademarks of T-HEAD Semiconductor Co.,Ltd. All other products or service names are the property of their respective owners.

Notice

The purchased products, services and features are stipulated by the contract made between T-HEAD and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied. The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Copyright © 2022 平头哥半导体有限公司，保留所有权利。

本文档的产权属于平头哥半导体有限公司(下称平头哥)。本文档仅能分布给:(i)拥有合法雇佣关系，并需要本文档的信息的平头哥员工，或(ii)非平头哥组织但拥有合法合作关系，并且其需要本文档的信息的合作方。对于本文档，禁止任何在专利、版权或商业秘密过程中，授予或暗示的可以使用该文档。在没有得到平头哥半导体有限公司的书面许可前，不得复制本文档的任何部分，传播、转录、储存在检索系统中或翻译成任何语言或计算机语言。

商标申明

平头哥的LOGO和其它所有商标归平头哥半导体有限公司所有，所有其它产品或服务名称归其所有者拥有。

注意

您购买的产品、服务或特性等应受平头哥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的购买或使用范围之内。除非合同另有约定，平头哥对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

平头哥半导体有限公司 T-HEAD Semiconductor Co.,LTD

地址:杭州市余杭区向往街1122号欧美金融城(EFC)英国中心西楼T6 43层 邮编: 311121

网址:www.T-head.cn

修订历史

日期	版本	新增修改内容	作者
20220420	v1.1	• 新增KDF工具	夏狼
20220328	v1.0	• 支持sectool工具说明	夏狼
20220311	v0.1	• 初版	夏狼

目录

[TOC]

1. 工具总览

工具名	说明	版本	仓库地址/附件
product	镜像签名工具	>= v1.0.23	随yoctool一起安装，参考yoctool用户手册： https://yuque.antfin-inc.com/occ/xyrz27/cywauid https://code.aone.alibaba-inc.com/thread/yoc_tools
efuse_utility	eFuse烧写工具	v1.2	仓库地址：git@gitlab.alibaba-inc.com:yocopen/efusehost.git 工具包：📎 efuse_host_v1.2.tar.zip 工具包md5um: 5d8257ce606232f21a69110e1654f192
	efuse key加密工具	v1.0	📎 efuse加解密脚本及使用方法说明 v1.0.7z
fastboot	镜像升级工具	v1.0	📎 fastboot v1.0.zip
sectool	imagesign.sh	v1.0	git@gitlab.alibaba-inc.com:thead-os-platform/sectool.git
	bin2ext4.sh		
	kdf_gen.exe		

2. 工具介绍

2.1 product

product用于系统镜像的签名，支持指定的密钥证书等功能。

功能命令：

sigx

说明：

对已生成好的单个镜像(或者公钥PEM文件)进行签名。

输入product sigx可查看帮助信息。

示例：

2级公钥的签名

- 国际算法例子：

```
product sigx keystore/pubkeyB.pem -pvk keystore/privatekeyA.pem -pubk  
keystore/pubkeyA.pem -ss RSA2048 -ds SHA256 -npubk keystore/pubkeyB.pem -nss RSA2048 -  
nds SHA256 -o sign_2nd_pubkey.bin
```

- 国密算法例子：

```
product sigx keystore_sm/pubkeyB.pem -pvk keystore_sm/privatekeyA.pem -pubk  
keystore_sm/pubkeyA.pem -ss SM2 -ds SM3 -npubk keystore_sm/pubkeyB.pem -nss SM2 -nds  
SM3 -o sign_2nd_pubkey.bin
```

镜像的签名(带有下级公钥)

- 国际算法例子：

```
product sigx iw.bin -pvk keystore/privatekeyB.pem -ss RSA2048 -ds SHA256 -npubk  
keystore/pubkeyC.pem -nss RSA2048 -nds SHA256 -iv 0 -ra 0xFFE0000000 -o sign_iw.bin
```

- 国密算法例子：

```
product sigx iw.bin -pvk keystore_sm/privatekeyB.pem -ss SM2 -ds SM3 -npubk  
keystore_sm/pubkeyC.pem -nss SM2 -nds SM3 -iv 0 -ra 0xFFE0000000 -o sign_iw.bin
```

镜像的加密签名(带有下级公钥)

- 国际算法例子：

```
product sigx iw.bin -pvk keystore/privatekeyB.pem -ss RSA2048 -ds SHA256 -npubk  
keystore/pubkeyC.pem -nss RSA2048 -nds SHA256 -ent AES_256_CBC -enk  
keystore/aes_256_cbc.key -iv 0 -ra 0xFFE0000000 -o sign_iw.bin
```

- 国密算法例子：

```
product sigx iw.bin -pvk keystore_sm/privatekeyB.pem -ss SM2 -ds SM3 -npubk  
keystore_sm/pubkeyC.pem -nss SM2 -nds SM3 -ent SM4_CBC -enk keystore_sm/sm4.key -iv 0  
-ra 0xFFE0000000 -o sign_iw.bin
```

镜像的签名(不带有下级公钥)

- 国际算法例子：

```
product sigx iw.bin -pvk keystore/privatekeyB.pem -ss RSA2048 -ds SHA256 -iv 0 -ra  
0xFFE0000000 -o sign_iw.bin
```

- 国密算法例子：

```
product sigx iw.bin -pvk keystore_sm/privatekeyB.pem -ss SM2 -ds SM3 -iv 0 -ra 0xFFE0000000  
-o sign_iw.bin
```

镜像的加密签名(不带有下级公钥)

- 国际算法例子：

```
product sigx iw.bin -pvk keystore/privatekeyB.pem -ss RSA2048 -ds SHA256 -ent AES_256_CBC  
-enk keystore/aes_256_cbc.key -iv 0 -ra 0xFFE0000000 -o sign_iw.bin
```

- 国密算法例子：

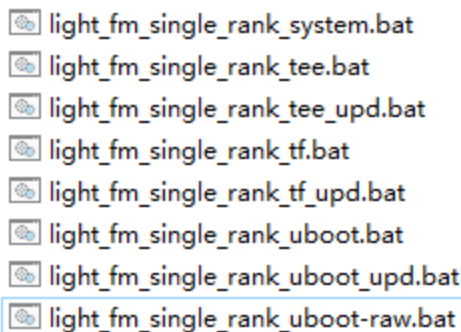
```
product sigx iw.bin -pvk keystore_sm/privatekeyB.pem -ss SM2 -ds SM3 -ent SM4_CBC -enk keystore_sm/sm4.key -iv 0 -ra 0xFFE0000000 -o sign_iw.bin
```

2.2 efuse utility

efuse utility是用于芯片efuse初次烧写，利用SPI Blaster来进行烧写，运行在Windows下的一个工具。

2.3 fastboot

fastboot是一种比recovery更底层的刷机模式（俗称引导模式）。就是使用USB数据线连接终端的一种刷机模式。我们利用fastboot进行系统镜像的更新。



- light_fm_single_rank_system.bat – 用于直接烧写所有系统镜像
- light_fm_single_rank_uboot.bat – 用于直接烧写uboot，不进行升级流程
- light_fm_single_rank_uboot_upd.bat – 用于更新uboot镜像，进行升级流程
- light_fm_single_rank_uboot_raw.bat – 用于更新烧写uboot镜像(uboot损坏)，不进行升级流程
- light_fm_single_rank_tee.bat – 用于直接烧写tee，不进行升级流程
- light_fm_single_rank_tee_upd.bat – 用于直接更新tee镜像，进行升级流程
- light_fm_single_rank_tf.bat – 用于直接烧写tf，不进行升级流程
- light_fm_single_rank_tf_upd.bat – 用于直接更新tf镜像，进行升级流程

注意：

Fastboot工具使用前需要安全驱动，请参考README指导安装。

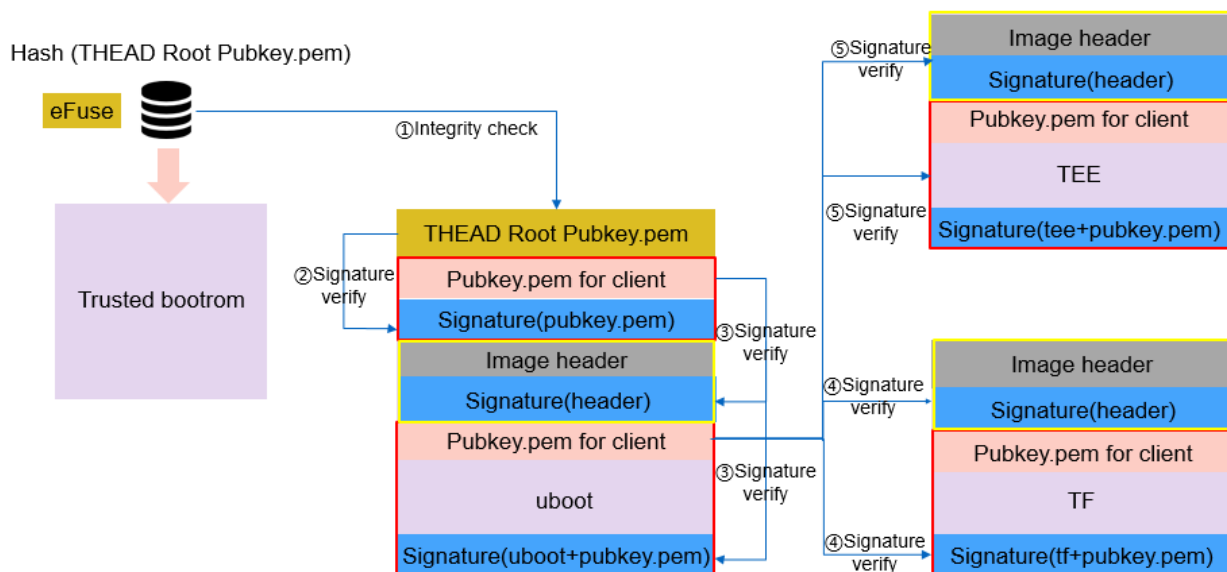
2.4 sectool

sectool用于镜像签名打包的工具，包括支持二进制文件打包成EXT4文件。其主要包括imagesign.sh和bin2ext4.sh脚本。

```
1 cxx194832@docker-ubuntu18:~/sectool$ ls
2 bin2ext4 imagesign.sh keystore keystore_sm README.md tee tf tool
  uboot
```

2.4.1 签名证书

Pubkey.pem for client表示二级签名私钥对应的公钥



在进行镜像签名前，可以在imagesign.sh脚本里通过修改以下两个变量来指定密钥证书。

- client_public_cert
- client_private_cert

注意：国际算法证书必须放在keystore文件夹下，国密算法证书必须放在keystore_sm文件夹下。以下证书用于uboot镜像签名，一般情况下由平头哥进行提供管理：

- thead_root_public_cert
- thead_root_private_cert
- thead_b1_public_cert
- thead_b1_private_cert

2.4.2 imagesign

imagesign工具是一个运行在shell下的一个脚本文件，其用于将一个二进制文件用指定的算法进行签名，同时还能指定镜像是否加密和版本号等信息，最后进行镜像内容和签名数据等信息打包，生成一个

签名文件。

▼

Bash | 复制代码

```
1  imagesign.sh utility version v1.0
2  The utility is designed for aim to help user generate new image file
3  with signature with desired sign scheme.
4
5  Usage:
6  imagesign.sh [ia/sm] [enc/nor] [tf/tee/u-boot] [ver]
7  signed algorithms
8      ia - international algorithm
9      sm - china algorithms
10 secure attribution
11     enc    - signed image with encryption
12     nor    - signed image without encryption
13 image file type
14     tf     - trust_firmware binary image
15     tee    - tee binary image
16     boot   - u-boot binary image
17 version
18     ver    - image version (x.y), eg 1.1, 2.1
```

2.4.2.1 生成UBOOT签名镜像

1. 将需要签名的u-boot-with-spl.bin复制到uboot文件夹里

注意：文件名必须是 **u-boot-with-spl.bin**，uboot文件夹名字不可以更改

▼

Bash | 复制代码

```
1  cxx194832@docker-ubuntu18:~/sectool$ ls uboot/
2  u-boot-with-spl.bin
```

2. 执行下面命令进行镜像签名


```

1  cxx194832@docker-ubuntu18:~/sectool$ ./imagesign.sh ia enc uboot 1.2
2  Dump all parameters from user input.
3  -----
4  Signed algorithm: ia
5  Secure attribution: enc
6  Image file: uboot
7  Image version: 1.2
8  -----
9  Enter into image sign process ...
10 Start uboot Image (1.2) signed with international algorithms with secure
    attr (enc)
11 sign tool path: ./tool/product
12 Original file: uboot/u-boot-with-spl.bin
13 Signed file: uboot/signed_ia_enc_u-boot-with-spl.bin
14 Image Version: 258
15 Relocate Addr: 0xFFE0000800
16 [2022-03-23 14:09:44] Sign a public key file.
17 [2022-03-23 14:09:44] rsa verify ok....
18 [2022-03-23 14:09:44] rsa verify ok....
19 [2022-03-23 14:09:44] Sign ok.
20 [2022-03-23 14:09:44] Sign a image file.
21 [2022-03-23 14:09:44] Sign ok.
22 Exit from image sign process ...
23

```

3. 查看生成的uboot签名镜像

```

1  cxx194832@docker-ubuntu18:~/sectool$ ls uboot/
2  signed_ia_enc_u-boot-with-spl.bin  signed_image_u-boot-with-spl.bin
3  signed_pubkey_u-boot-with-spl.bin  u-boot-with-spl.bin

```

生成的 `signed_ia_enc_u-boot-with-spl.bin` 位于uboot文件夹下。

注意：

- uboot签名镜像由 `signed_pubkey_u-boot-with-spl.bin` 和 `signed_image_u-boot-with-spl.bin` 两部分签名镜像组成。
- `signed_pubkey_u-boot-with-spl.bin` 由平头哥管理
- `signed_image_u-boot-with-spl.bin` 用于镜像升级，但签名镜像的私钥必须和 `signed_`

`pubkey_u-boot-with-spl.bin` 了用的公钥配对。

- 使用的时候必须将 `signed_ia_enc_u-boot-with-spl.bin` 改名为 `u-boot-with-spl.bin`

2.4.2.2 生成TF签名镜像

1. 将需要签名的`trust_firmware.bin`复制到tf文件夹里

注意：文件名必须是 `trust_firmware.bin`，tf文件夹名字不可以更改

```
▼ Bash | 复制代码
1 cxx194832@docker-ubuntu18:~/sectool$ ls tf
2 trust_firmware.bin
```

2. 执行下面命令进行镜像签名

```
▼ Bash | 复制代码
1 cxx194832@docker-ubuntu18:~/sectool$ ./imagesign.sh ia nor tf 1.2
2 Dump all parameters from user input.
3 -----
4 Signed algorithm: ia
5 Secure attribution: nor
6 Image file: tf
7 Image version: 1.2
8 -----
9 Enter into image sign process ...
10 Start tf Image (1.2) signed with international algorithms with secure
    attr (nor)
11 sign tool path: ./tool/product
12 Original file: tf/trust_firmware.bin
13 Signed file: tf/signed_ia_nor_trust_firmware.bin
14 Image Version: 258
15 Relocate Addr: 0x0
16 [2022-03-23 14:12:36] Sign a image file.
17 [2022-03-23 14:12:36] Sign ok.
18 Exit from image sign process ...
```

3. 查看生成的TF签名镜像

```
1 cxx194832@docker-ubuntu18:~/sectool$ ls tf
2 signed_ia_nor_trust_firmware.bin trust_firmware.bin
```

生成的 `signed_ia_nor_trust_firmware.bin` 位于tf文件夹下。

注意：

- 使用的时候必须将 `signed_ia_nor_trust_firmware.bin` 改名为 `trust_firmware.bin`

2.4.2.3 生成TEE签名镜像

1. 将需要签名的tee.bin复制到tee文件夹里

注意：文件名必须是 `trust_firmware.bin` ， tf文件夹名字不可以更改

```
1 cxx194832@docker-ubuntu18:~/sectool$ ls tee
2 tee.bin
```

2. 执行下面命令进行镜像国密算法签名

```
1 cxx194832@docker-ubuntu18:~/sectool$ ./imagesign.sh ia nor tee 1.2
2 Dump all parameters from user input.
3 -----
4 Signed algorithm: ia
5 Secure attribution: nor
6 Image file: tee
7 Image version: 1.2
8 -----
9 Enter into image sign process ...
10 Start tee Image (1.2) signed with international algorithms with secure
    attr (nor)
11 sign tool path: ./tool/product
12 Original file: tee/tee.bin
13 Signed file: tee/signed_ia_nor_tee.bin
14 Image Version: 258
15 Relocate Addr: 0xff000000
16 [2022-03-23 14:13:31] Sign a image file.
17 [2022-03-23 14:13:31] Sign ok.
18 Exit from image sign process ...
19
```

3. 查看生成的TEE签名镜像

```
1 cxx194832@docker-ubuntu18:~/sectool$ ls tee
2 signed_sm_nor_tee.bin tee.bin
```

生成的 `signed_sm_nor_tee.bin` 位于tee文件夹下。

注意：

- 使用的时候必须将 `signed_sm_nor_tee.bin` 改名为 `tee.bin`

2.4.3 bin2ext4

bin2ext4工具用于将一个文件打包生成EXT4文件。

注意：在默认情况下，打包的EXT4文件大小是8M，如果文件真实大小大于8M，需要调整成合适的值。



Bash

复制代码

```
1 dd if=/dev/zero of=$1 bs=1M count=8
```

2.4.3.1 生成uboot.EXT4

1. 将 `u-boot-with-spl.bin` 复制到bin2ext4文件下



Bash

复制代码

```
1 cxx194832@docker-ubuntu18:~/sectool/bin2ext4$ ls
2 bin2ext4.sh u-boot-with-spl.bin
```

2. 执行下面命令生成 `uboot.ext4`



Bash

复制代码

```
1 cxx194832@docker-ubuntu18:~/sectool/bin2ext4$ ./bin2ext4.sh uboot.ext4 u-
  boot-with-spl.bin
2 bin2ext4 utility version v1.0
3
4 1+0 records in
5 1+0 records out
6 1048576 bytes (1.0 MB, 1.0 MiB) copied, 0.00185016 s, 567 MB/s
7 mke2fs 1.44.1 (24-Mar-2018)
8
9 Filesystem too small for a journal
10 Discarding device blocks: done
11 Creating filesystem with 1024 1k blocks and 128 inodes
12
13 Allocating group tables: done
14 Writing inode tables: done
15 Writing superblocks and filesystem accounting information: done
```

3. 查看生成的 `uboot.ext4` 文件

Bash | 复制代码

```
1 cxx194832@docker-ubuntu18:~/sectool/bin2ext4$ ls
2 bin2ext4.sh u-boot-with-spl.bin uboot.ext4
```

2.4.3.2 生成TF.EXT4

1. 将 `trust_firmware.bin` 复制到bin2ext4文件下

Bash | 复制代码

```
1 cxx194832@docker-ubuntu18:~/sectool/bin2ext4$ ls
2 bin2ext4.sh trust_firmware.bin
```

2. 执行下面命令生成tf.ext4

Bash | 复制代码

```
1 cxx194832@docker-ubuntu18:~/sectool/bin2ext4$ ./bin2ext4.sh tf.ext4
  trust_firmware.bin
2 bin2ext4 utility version v1.0
3
4 1+0 records in
5 1+0 records out
6 1048576 bytes (1.0 MB, 1.0 MiB) copied, 0.00112569 s, 931 MB/s
7 mke2fs 1.44.1 (24-Mar-2018)
8
9 Filesystem too small for a journal
10 Discarding device blocks: done
11 Creating filesystem with 1024 1k blocks and 128 inodes
12
13 Allocating group tables: done
14 Writing inode tables: done
15 Writing superblocks and filesystem accounting information: done
```

3. 查看生成的tf.ext4文件

Bash | 复制代码

```
1 cxx194832@docker-ubuntu18:~/sectool/bin2ext4$ ls
2 bin2ext4.sh trust_firmware.bin tf.ext4
```

2.4.3.3 生成TEE.EXT4

1. 将 `tee.bin` 复制到bin2ext4文件下

Bash | 复制代码

```
1 cxx194832@docker-ubuntu18:~/sectool/bin2ext4$ ls
2 bin2ext4.sh tee.bin
```

2. 执行下面命令生成tee.ext4

Bash | 复制代码

```
1 cxx194832@docker-ubuntu18:~/sectool/bin2ext4$ ./bin2ext4.sh tf.ext4
tee.bin
2 bin2ext4 utility version v1.0
3
4 1+0 records in
5 1+0 records out
6 1048576 bytes (1.0 MB, 1.0 MiB) copied, 0.00112569 s, 931 MB/s
7 mke2fs 1.44.1 (24-Mar-2018)
8
9 Filesystem too small for a journal
10 Discarding device blocks: done
11 Creating filesystem with 1024 1k blocks and 128 inodes
12
13 Allocating group tables: done
14 Writing inode tables: done
15 Writing superblocks and filesystem accounting information: done
```

3. 查看生成的tee.ext4文件

```
1 cxx194832@docker-ubuntu18:~/sectool/bin2ext4$ ls
2 bin2ext4.sh tee.bin tee.ext4
```

2.4.4 KDF_GEN

KDF_GEN工具根据输入的CV_ROOTKEY和CV_COMMONKEY来派生KDF密钥，存在kdf_derived_key.dat，同时产生测试向量存在kdf_test_vector.dat里。

1. 将CV_ROOTKEY写入cv_rootkey.dat文件
2. 将CV_COMMONKEY写入cv_commonkey.dat文件
3. 执行key_gen.elf
4. 查看kdf_derived_key.dat和kdf_test_vector.dat

其中kdf_derived_key.dat包含所有预期生成的派生密钥，kdf_test_vector.dat包含生成派生密钥的测试向量。

3. 用户手册

文档名	文档链接
product用户手册	product用户手册_v1.2.pdf
fastboot用户手册	USB-Fastboot用户手册_v1.0.0.pdf
efuse上位机用户手册	eFuse上位机用户手册_v1.0.0.pdf