

君正®

SecurityBoot 工具说明文档

Date: Feb 2023



北京君正集成电路股份有限公司
Ingenic Semiconductor Co., Ltd.

Copyright © 2005-2023 Ingenic Semiconductor Co. Ltd. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Ingenic Semiconductor Co. Ltd.

Trademarks and Permissions



、 Ingenic and Ingenic icons are trademarks of Ingenic Semiconductor Co.Ltd. All other trademarks and trade names mentioned in this document are the property of their respective holders.

Disclaimer

All the deliverables and data in this folder serve only as a reference for customer development. Please read through this disclaimer carefully before you use the deliverables and data in this folder. You may use the deliverables in this folder or not. However, by using the deliverables and data in this folder, you agree to accept all the content in this disclaimer unconditional and irrevocable. If you do not find the content in this disclaimer reasonable, you shall not use the deliverables and data in this folder. The deliverables and data in this folder are provided "AS IS" without representations, guarantees or warranties of any kind (either express or implied). To the maximum extent permitted by law, Ingenic Semiconductor Co., Ltd (Ingenic) provides the deliverables and data in this folder without implied representations, guarantees or warranties, including but not limited to implied representations, guarantees and warranties of merchantability, non-infringement, or fitness for a particular purpose. Deviation of the data provided in this folder may exist under different test environments.

Ingenic takes no liability or legal responsibility for any design and development error, incident, negligence, infringement, and loss (including but not limited to any direct, indirect, consequential, or incidental loss) caused by the use of data in this folder. Users shall be responsible for all risks and consequences caused by the use of data in this folder.

Ingenic Semiconductor Co., Ltd.

Add: Ingenic Headquarters, East Bldg. 14, Courtyard #10, Xibeiwang East Road, Haidian Dist., Beijing, China

Tel: **(86-10)56345000**

Fax: **(86-10)56345001**

<http://www.ingenic.cn>

目录

1 SecurityBoot 简述.....	4
2 安全启动配置.....	4
3 工具使用流程.....	4
4 工具介绍.....	5
4.1 密钥工具.....	6
4.2 签名工具.....	6
4.2.1 burner bin.....	6
4.2.2 spl bin.....	7
4.2.3 uboot_with_spl bin.....	7
4.2.4 kernel.....	8
4.2.5 other file.....	9
4.3 烧录工具.....	9
5 安全建议.....	11

版本历史

日期	版本	描述
Feb, 2023	2.0	发布第二版
Apr, 2017	1.0	发布第一版

1 SecurityBoot 简述

Security Boot 主要为了保护客户厂商的软件程序不被其对手复制，包括整体抄板或者抄袭某个应用程序。与此同时，还要允许终端客户进行软件升级或重新烧录软件程序。

防止代码复制是通过对代码进行加密的方式实现，可以使用 AES 加密，密钥可以由用户提供的 User-Key 或芯片内部的 Chip-Key（随机数）。

防止加密的代码被解密，我们将密钥存放在芯片的 OTP 中，只有芯片中的 SC-ROM 代码可以访问。可以通过调用 SC-ROM 接口使用密码进行加解密，而软件不直接访问密码本身，防止密码泄露。

2 安全启动配置

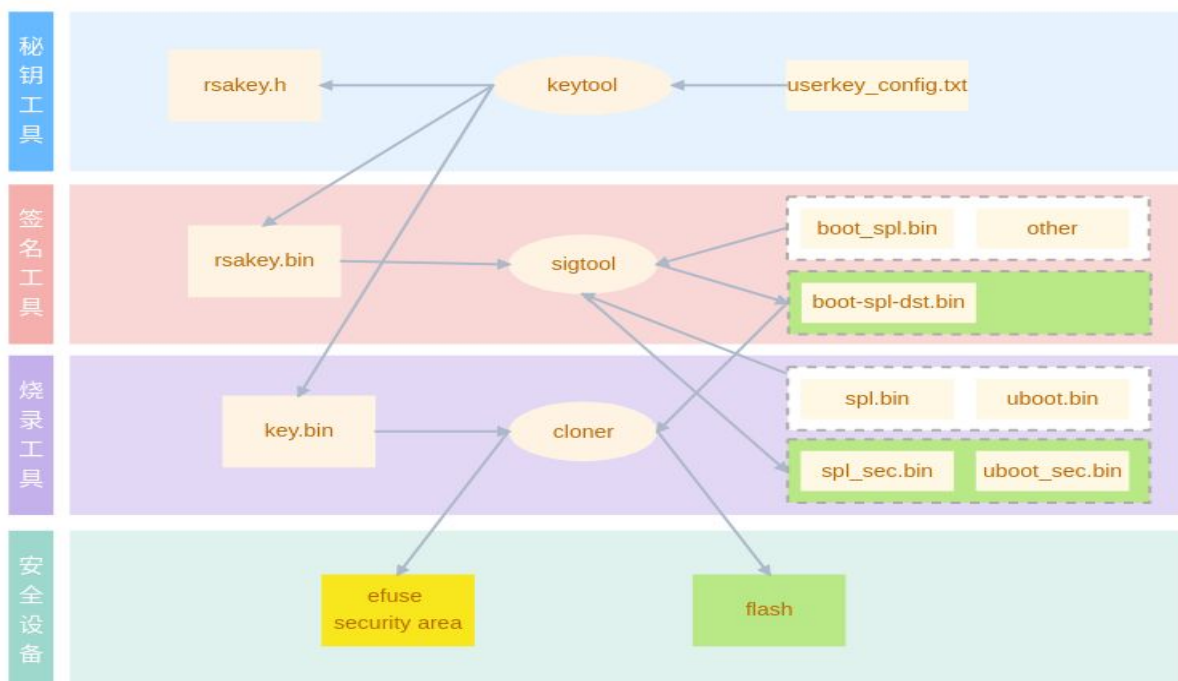
启用安全启动功能需要修改 UBOOT 板级配置文件，路径如下：

```
u-boot/include/configs/{board}.h
```

添加如下配置：

```
#define CONFIG_JZ_SCBOOT
#define CONFIG_JZ_SECURE_SUPPORT
```

3 工具使用流程

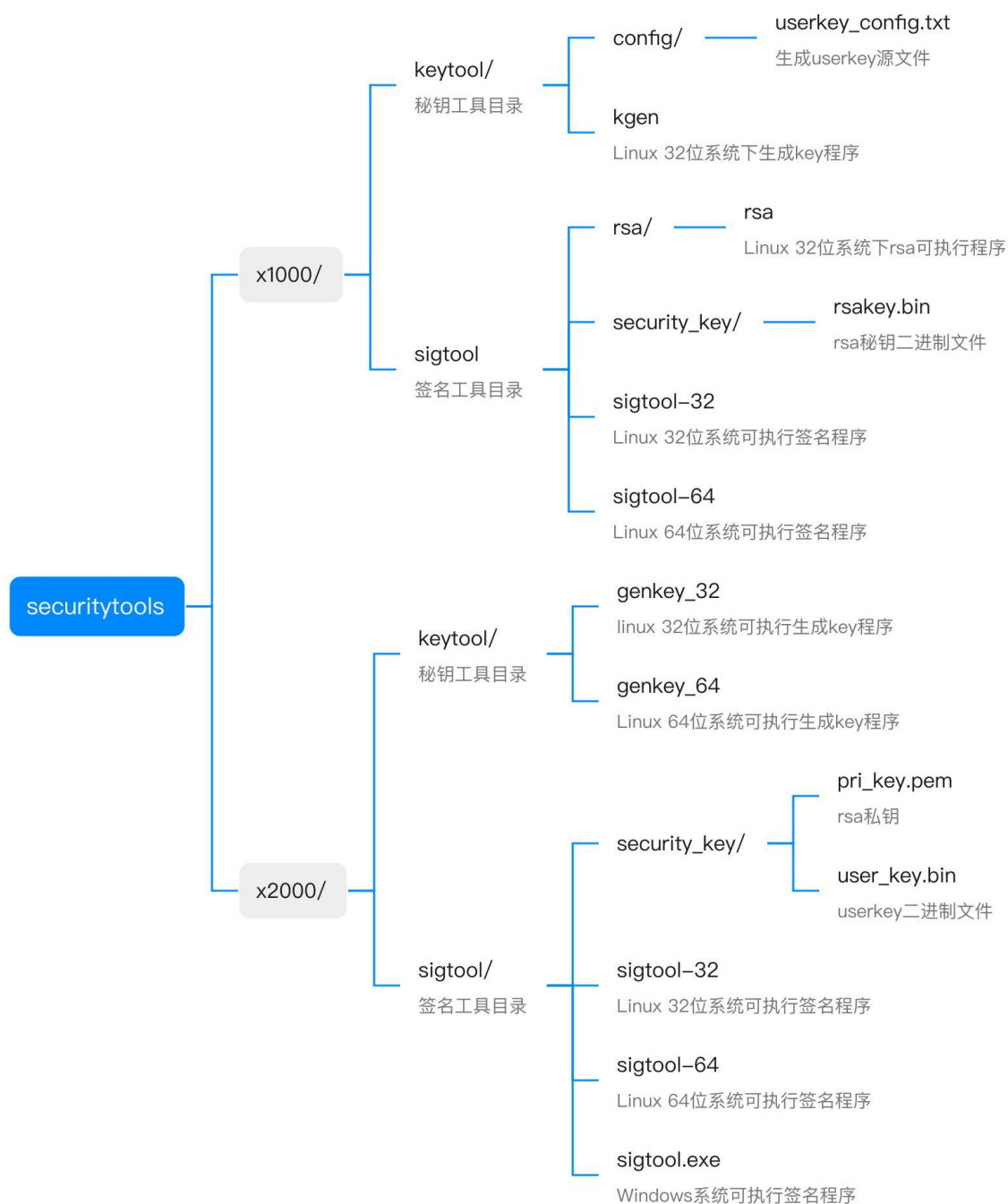


4 工具介绍

在使用 SecurityBoot 功能时，需要用到以下工具：

名称	功能
密钥工具(keytool)	用于生成 RSA-Key、User-Key 提供给签名工具、烧录工具和 SC-ROM 使用。
签名工具(sigtool)	使用生成的 RSA 私钥、User-Key 对数据进行加密和签名，并将加密信息填充到文件起始的 2048 字节中。
烧录工具(cloner)	烧录 RSA 公钥的 HASH 值、User-Key 和加密后的软件程序。

在烧录工具 securitytools 目录下包含密钥和签名工具。目录结构如下所示：



4.1 密钥工具

用于生成 RAS-Key、User-Key 提供给签名工具和烧录工具使用。目前 ku(公钥)的长度支持 17-31 字长，在不失密钥的安全性，同时兼顾缩短启动时间的情况下，17 个字长为推荐配置，如果需要更高的密钥安全性，可以选择 31 个字长。

使用方法: kgen <kn 长度> <ku 长度>

例如执行 “./kgen 31 17” 命令后，会在当前路径下生成如下几个文件：

key.bin	包含公钥、userkey，放到烧录工具 security 目录下。
rsakey.bin	包含所有 key 信息，放到签名工具 security_key 目录下。
rsakey.h	所有 key 信息以数组方式存放在此文件中。

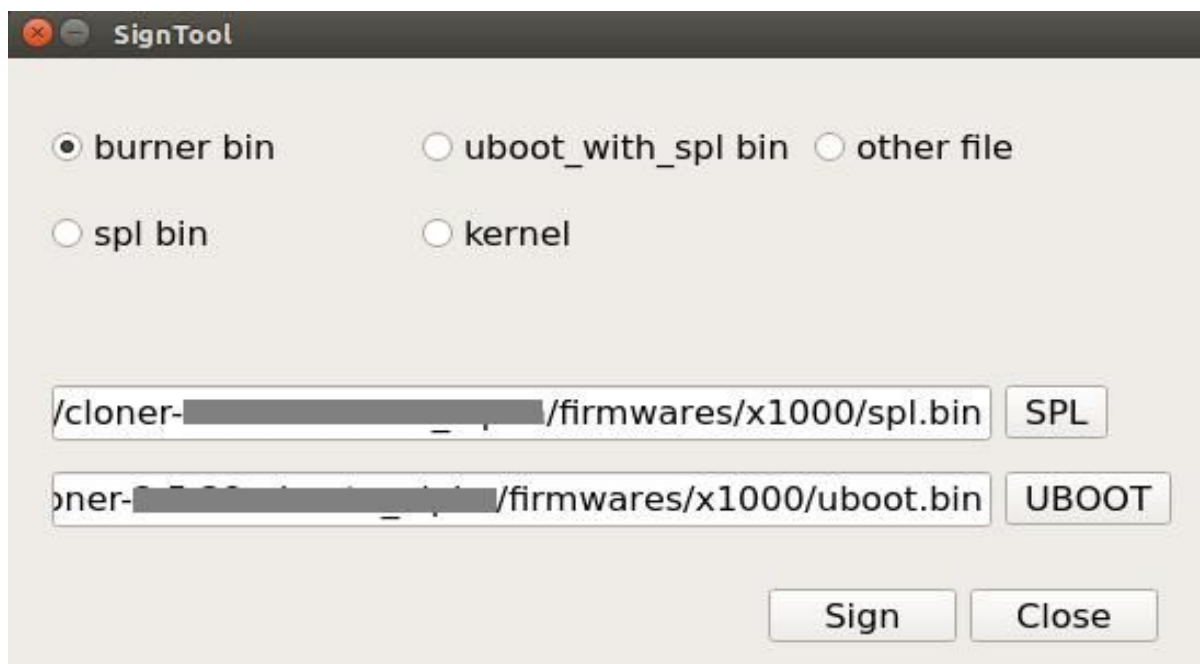
注意：

1. userkey 是 config/userkey_config.txt 文件的 md5 值，生成 key 之前修改此文件内容，以保证 userkey 的唯一性。
2. 必须同步更新烧录工具和签名工具所使用的 key 文件，保证 key 的一致性。
3. 请妥善保存好您所使用的 key 文件。

4.2 签名工具

用于烧录固件、应用程序、配置文件的签名和加密保护。

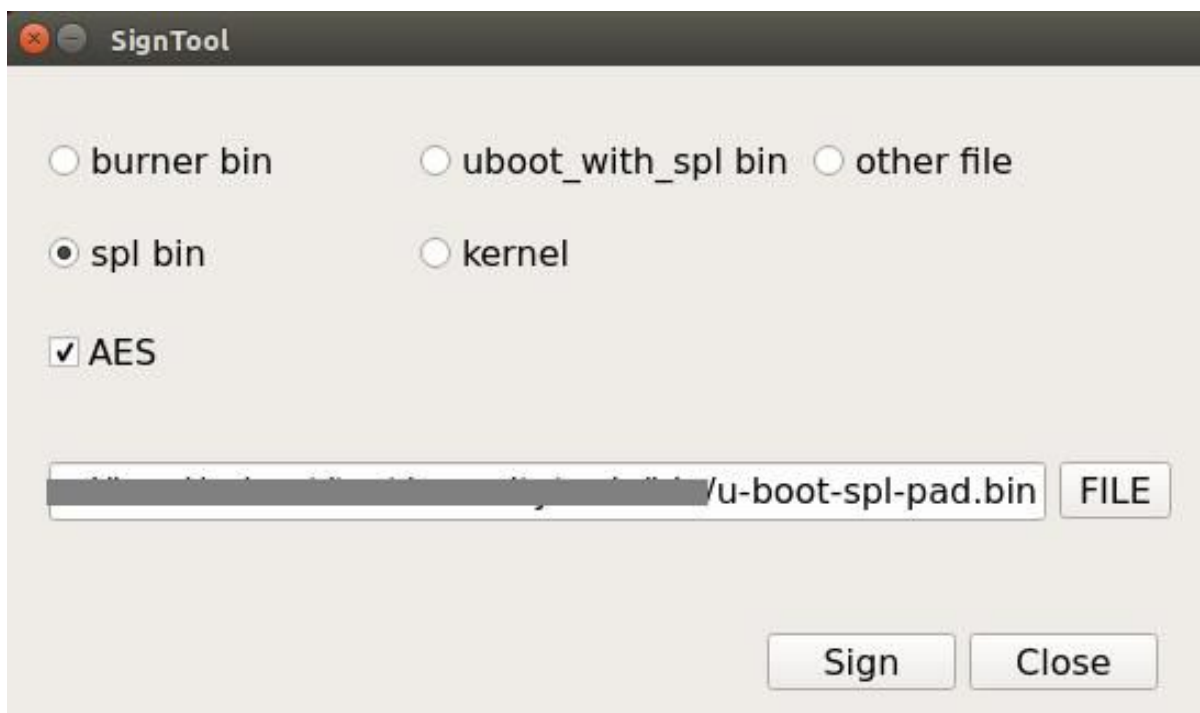
4.2.1 burner bin



烧录固件签名和加密操作步骤：

步骤	描述
1	选中“burner bin”选项
2	点击“SPL”按钮，选择烧录工具目录下的 firmwares/x1000/spl.bin 文件。
3	点击“UBOOT”按钮，选择烧录工具目录下的 firmwares/x1000/uboot.bin 文件。
4	点击“Sign”按钮，签名成功后在烧录工具 firmwares/x1000/ 目录下生成 spl_sec.bin 和 uboot_sec.bin 文件。

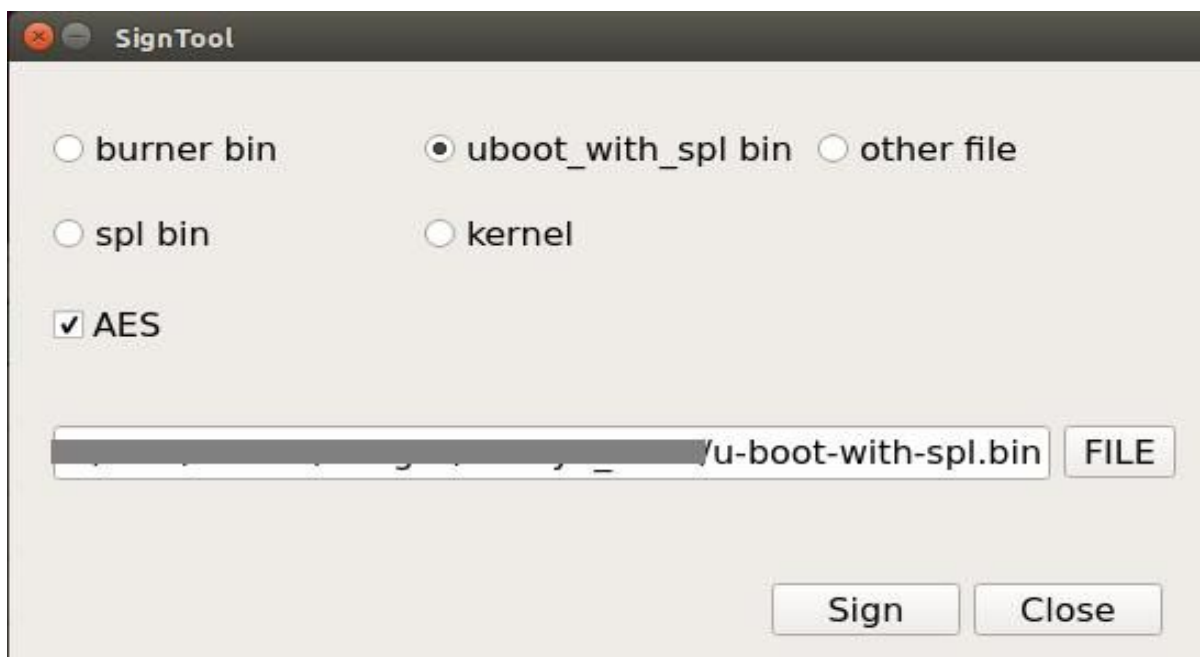
4.2.2 spl bin



启动固件 SPL 签名和加密操作步骤:

步骤	描述
1	选中“spl bin”选项。
2	选中“AES”选项后数据加密，否则不加密。
3	点击“FILE”按钮，选择 uboot 源码编译生成的 spl/u-boot-spl-pad.bin 文件所在路径。

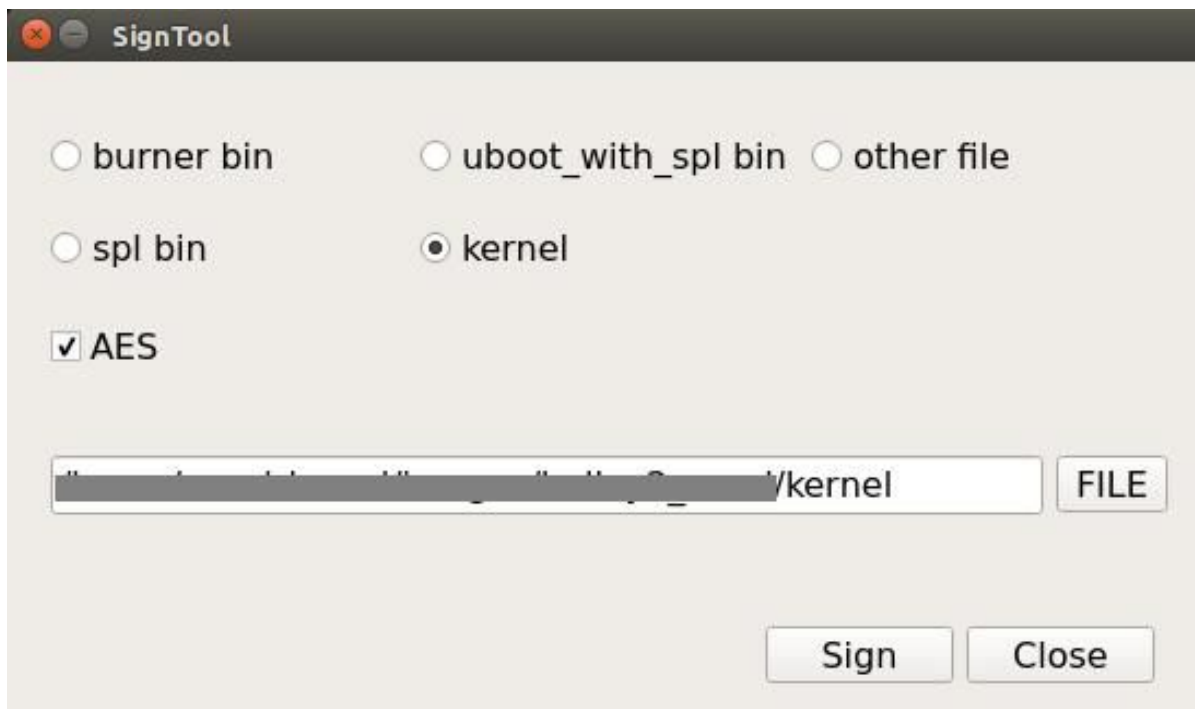
4.2.3 u-boot_with_spl bin



spl 和 uboot 合并后的启动固件签名和加密步骤：

步骤	描述
1	选中“uboot-wiht-spl bin”选项。
2	选中“AES”选项后数据加密，否则不加密。
3	点击“FILE”按钮，选择 uboot 源码编译生成的 u-boot-with-spl.bin 文件所在路径。
4	点即“Sign”按钮，签名成功后在 u-boot-with-spl.bin 所在目录下生成 u-boot-with-spl-dst.bin 文件。

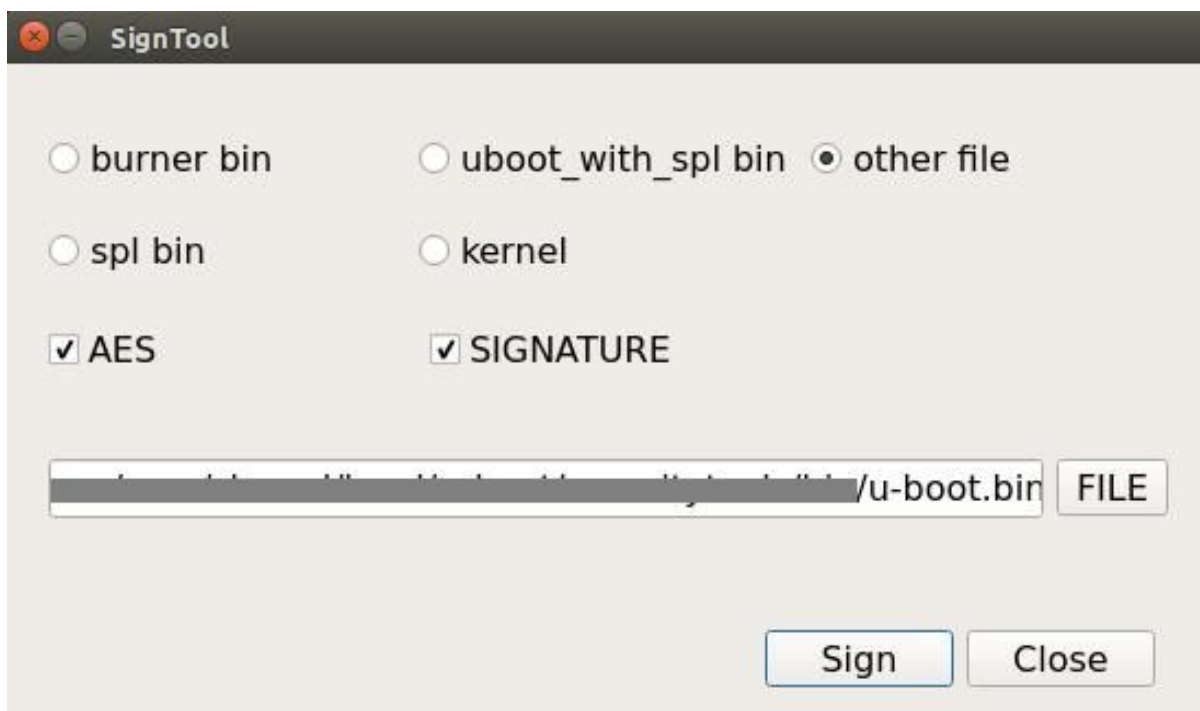
4.2.4 kernel



启动固件 kernel 签名和加密步骤：

步骤	描述
1	选中“kernel”选项。
2	选中“AES”选项后数据加密，否则不加密。
3	点击“FILE”按钮，选择 kernel 源码编译生成的 uImage 或者 xImage 文件。
4	点即“Sign”按钮，签名成功后在 kernel 所在目录下生成 kernel-dst 文件。

4.2.5 other file

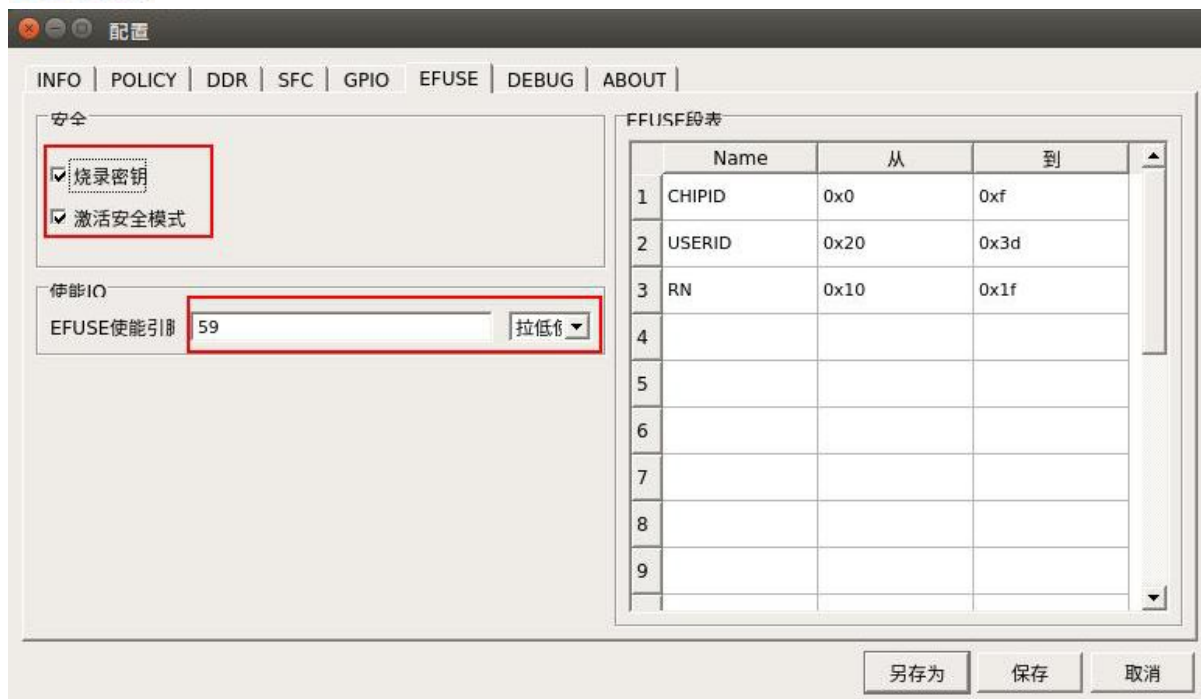


其他文件或者单独的 UBOOT 文件签名和加密步骤：

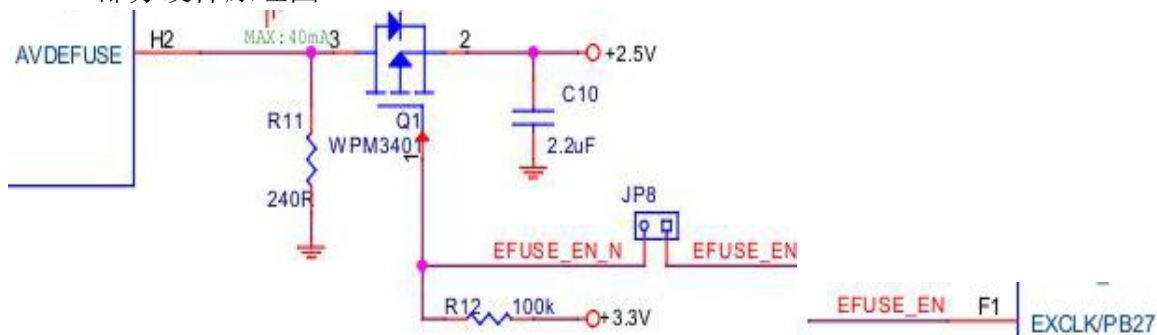
步骤	描述
1	选中“other file”选项。
2	选中“AES”选项后数据加密，否则不加密。
3	选中“SIGNATURE”选项后使用 RSA 生成数据签名，否则不签名。
4	点击“FILE”按钮，选择 kernel 源码编译生成的 uImage 或者 xImage 文件。
5	点击“Sign”按钮，签名成功后在 kernel 所在目录下生成 kernel-dst 文件。

4.3 烧录工具





EFUSE 部分硬件原理图



烧录工具配置步骤:

步骤	描述
1	运行 cloner 程序
2	点击“配置”按钮，选择 X1000 平台和相应板级配置文件。
3	选中“强制重启”选项，烧录激活安全模式后需要重启设备。
4	点击“EFUSE”标签，切换到 EFUSE 配置页面。
5	选中“烧录密钥”选项，烧录密钥到 EFUSE 中。
6	选中“激活安全模式”选项，置 SCB00T 使能位。
7	输入 EFUSE 使能引脚号，例如：EFUSE_EN 为 PB27，引脚号为 59。注：(32+27)
8	点击“POLICY”标签，选择签名和加密后的安全启动固件路径。
9	点击“保存”按钮，关闭配置窗口。
10	点击“开始”按钮，开始烧录密钥、激活安全模式、重启设备、烧录安全固件。

注意：EFUSE 只能写一次，被激活安全模式的芯片无法恢复正常模式，烧录非安全固件无法启动！

5 安全建议

1. 密钥要有最高的保密级别，不出厂商的安全部门。
2. 工厂生产或用网络上系统升级，固件需要签名加密。
3. 设置 EFUSE 中的 DISJTAG 位，降低破解风险。
4. 厂商不要保留 ChipKey，降低破解风险。
5. 安全固件解密完成后及时清理内存和复位寄存器。