

安全启动使用指南

文档版本 V1.2

发布日期 2025-08-20

前言

概述

本文档主要介绍 XM7606V13、XM7206V11A 安全启动的使用方法，主要内容包括：安全启动介绍、安全镜像生成步骤以及安全启动相关的 OTP 烧写说明。



说明

未经特殊说明，以 XM7606V13 指代 XM76 系列所有芯片。

以 XM2606V11A 指代 XM7206V1 系列所有芯片。

读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

修订日期	版本	修订说明
2024-12-21	V1.0	DI 版本发布。
2025-06-04	V1.1	支持 XM7206V11A 平台。
2025-08-20	V1.2	修订 XM7206V11A OTP 地址。

目 录

1 安全启动介绍	1
1.1 普通安全 boot 镜像结构.....	2
1.2 加密安全 boot 镜像结构.....	3
1.3 Params area.....	4
1.4 安全启动流程.....	5
1.5 安全启动源代码目录说明.....	6
1.5.1 XM7606V13 目录结构.....	6
1.5.2 XM7206V11A 目录结构	7
2 安全镜像生成	8
2.1 安全 boot 镜像生成步骤.....	8
2.2 密钥文件介绍.....	9
3 OTP 烧写介绍.....	10
3.1 操作步骤	10
3.1.1 XM7606V13 OTP 区域烧写	10
3.1.2 XM7206V11A OTP 区域烧写	12

插图目录

图 1-1 普通安全 u-boot 镜像结构	2
图 1-2 加密安全 u-boot 镜像结构	3
图 1-3 Params area	4
图 1-4 安全启动流程	5

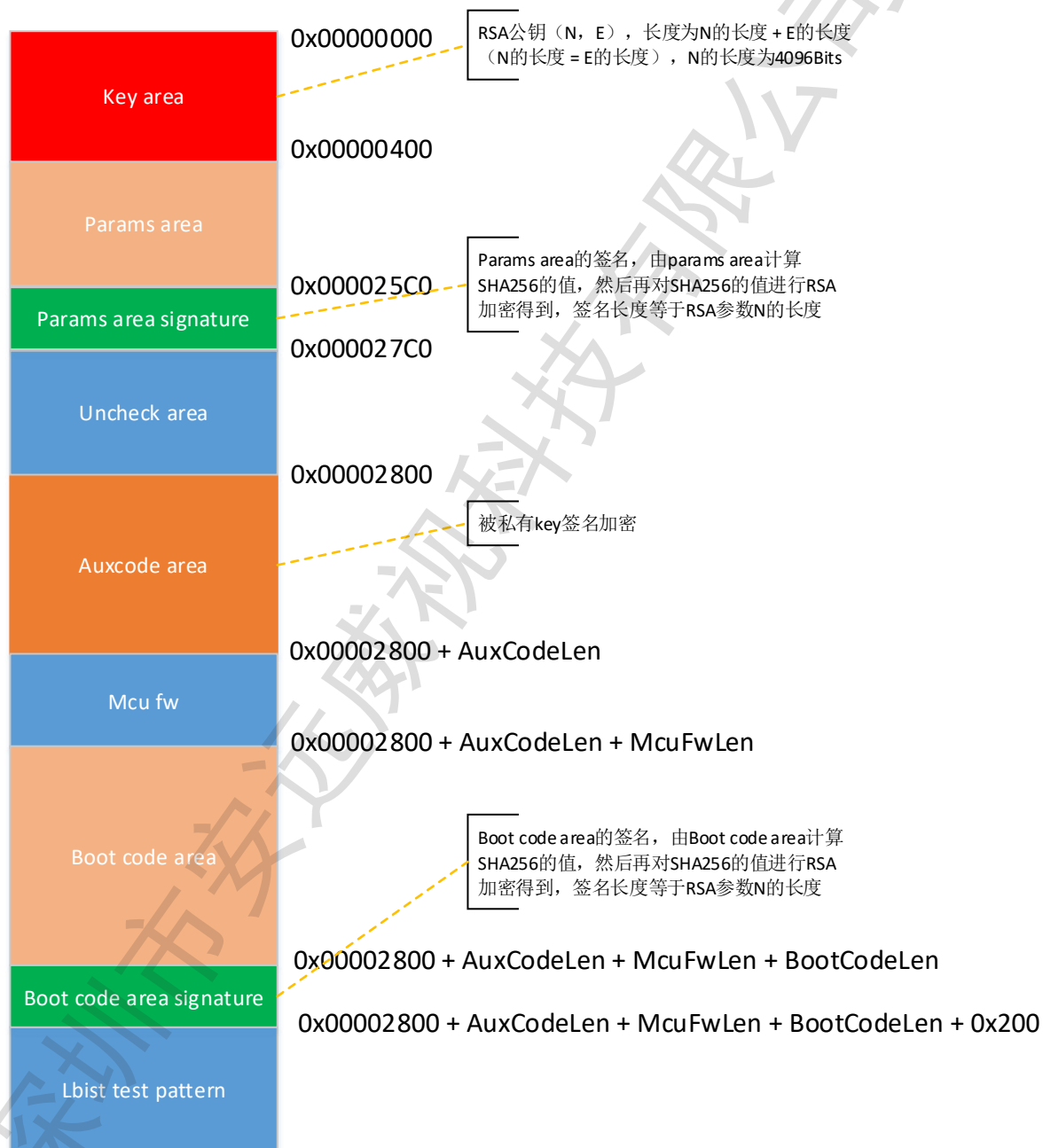
1 安全启动介绍

XM7606V13、XM7206V11A 支持普通安全 boot 启动和加密安全 boot 启动，其差异在于普通安全 boot 镜像中，u-boot.bin 的 boot code area 是明文；加密的安全 boot 镜像中，u-boot.bin 的 boot code area 是密文。

1.1 普通安全 boot 镜像结构

XM7606V13、XM7206V11A 普通安全 u-boot 镜像结构如图 1-1 所示。

图1-1 普通安全 u-boot 镜像结构

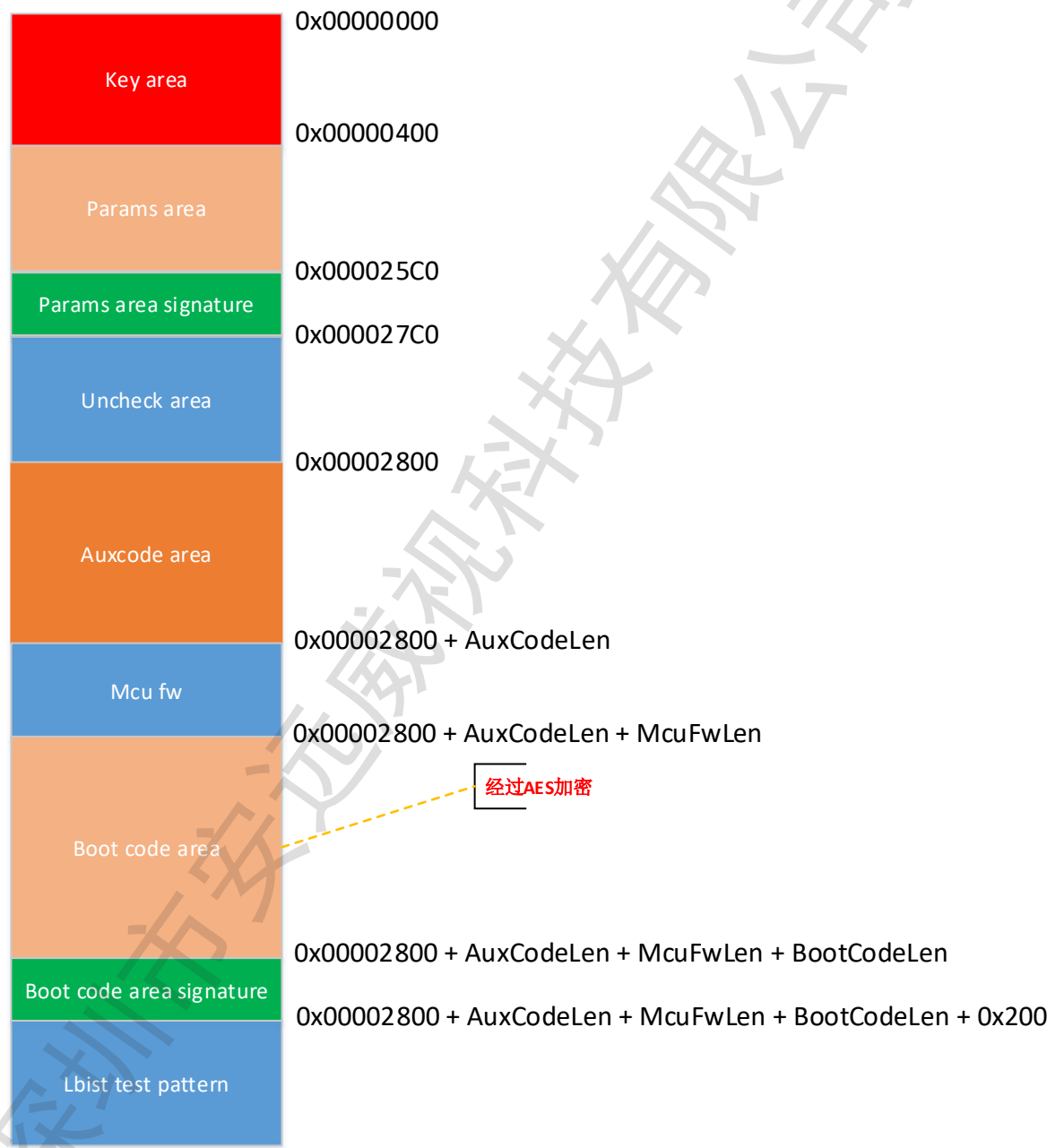


普通安全 uboot 镜像 Auxcode area 是密文, 其它区域都是明文。

1.2 加密安全 boot 镜像结构

XM7606V13、XM7206V11A 加密安全 u-boot 镜像结构如图 1-2 所示。

图1-2 加密安全 u-boot 镜像结构



加密安全 uboot 镜像 Auxcode area 和 Bootcode area 是密文，其它区域都是明文。

1.3 Params area

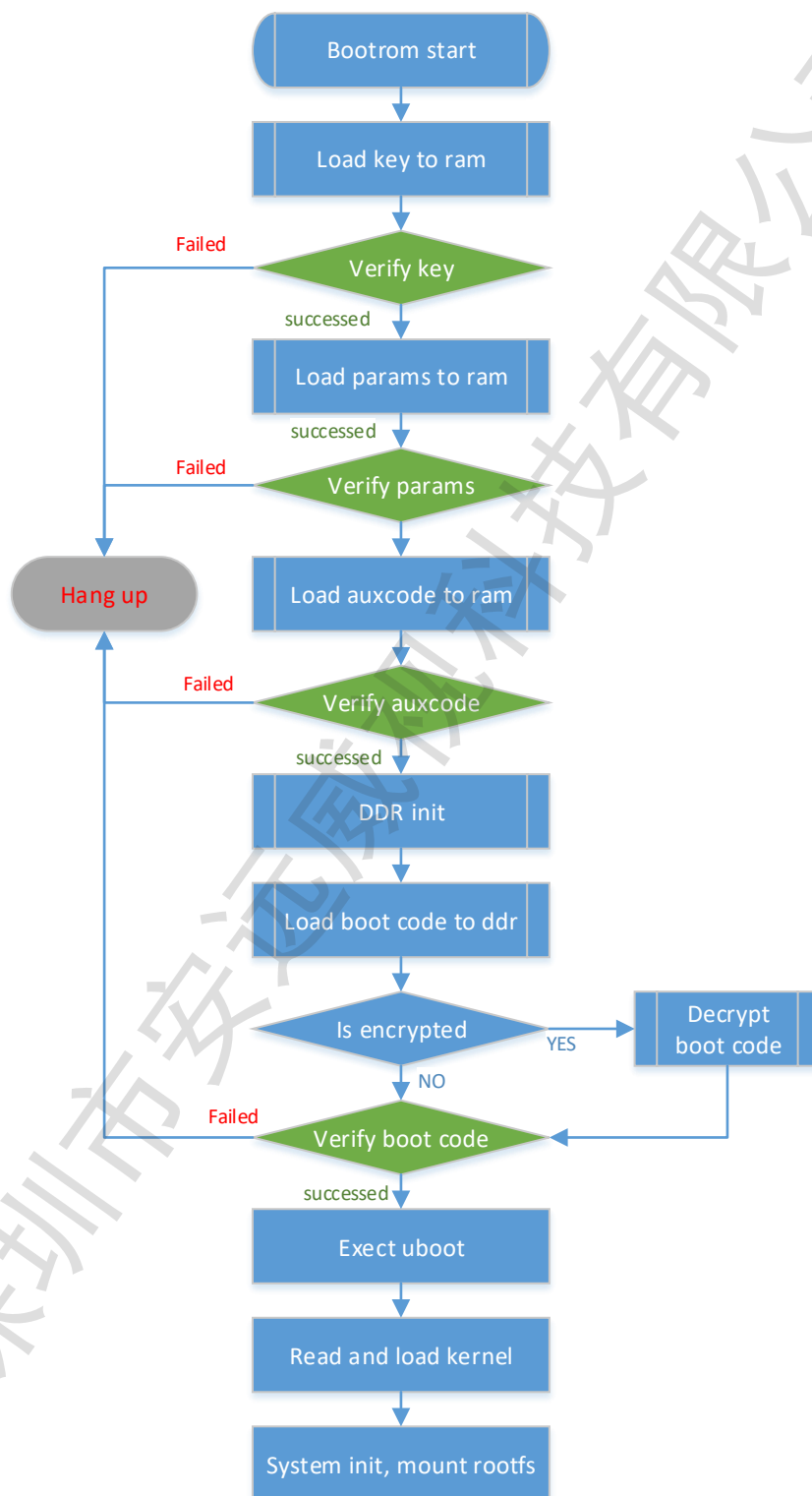
Params area 中包含 uboot 各域的长度信息、加密标志以及加密 IV 等信息。

图1-3 Params area

	3	7	B	F
0x0400	AUXCODE_AREA_LEN	BOOT_CODE_AREA_LEN	TOTAL_BOOT_AREA_LEN	PARAMS_FORM_LEN
0x0410	BOOT_ENC_FLAG	BOOT_ENTRY_PIONT	AUXCODE_AES_IV	
0x0420	AUXCODE_AES_IV		BOOT_AES_IV	
0x0430	BOOT_AES_IV		PARAMS_FORM(REG)	
	PARAMS_FORM(REG)			
0x2030	PARAMS_FORM(REG)		LOAD_MCU_FW_FLAG	MCU_FW_AREA_LEN
0x2040	LBIST_TEST_PATTERN_OFF	LBIST_TEST_PATTERN_LEN	LBIST_TEST_PATTERN_HASH	
0x2050	LBIST_TEST_PATTERN_HASH			
0x2060	LBIST_TEST_PATTERN_HASH		AUXCODE_ALG_FLAG	00000000

1.4 安全启动流程

图1-4 安全启动流程



步骤 1 上电, bootrom 启动。

步骤 2 Bootrom 加载 key area 到 ram, 计算 key area 的 sha256 值, 并和 OTP 中 RSA 公钥哈希值做对比校验。成功将继续执行, 失败则挂起。

步骤 3 Bootrom 加载 params area 到 ram, 并使用 key area 的 RSA 公钥验签 params area。成功将继续执行, 失败则挂起。

步骤 4 Bootrom 加载 auxcode area 到 ram, 并使用固化在再 bootrom 中的私有 key 解密、校验 auxcode。成功将继续执行, 失败则挂起。

步骤 5 Bootrom 执行 auxcode 中 ddr init 固件, 完成 ddr 初始化。

步骤 6 Bootrom 加载 boot code area 到 ddr。若 boot code area 已加密, 则先使用 OTP 中的 AES key 解密, 再使用 key area 的 RSA 公钥验签。若 boot code area 是明文, 则直接使用 key area 的 RSA 公钥验签。成功则继续执行, 失败则挂起。

步骤 7 执行 u-boot。

步骤 8 Uboot 加载并运行 kernel。

步骤 9 Kernel 启动, 挂在文件系统。

----结束

1.5 安全启动源代码目录说明

安全启动源代码目录为 sdk/source/bootloader/secureboot\$, 详细结构如下:

XM7606V13 目录结构

```
.
├── create_secure_boot_v3.sh ----- 生成安全镜像脚本
├── Makefile ----- 安全启动发布包 Makefile
├── readme.txt ----- 安全启动发布包使用说明
└── rsa4096pem.sh ----- 生成长度为 4096bit 密钥脚本
```

XM7206V11A 目录结构

- └─ **create_secure_boot_v4.sh** ----- 生成安全镜像脚本
- └─ Makefile ----- 安全启动发布包 Makefile
- └─ readme.txt ----- 安全启动发布包使用说明
- └─ rsa4096pem.sh ----- 生成长度为 4096bit 密钥脚本

2 安全镜像生成

2.1 安全 boot 镜像生成步骤

步骤 1 生成非安全 boot 镜像。

参考《SDK 开发环境用户指南》中的描述，编译出基础 SDK 镜像，其中即包括非安全 boot 镜像。

步骤 2 将非安全 boot 拷贝到 secureboot 目录。

将步骤一生成的非安全 boot 镜像拷贝至 source/bootloader/secureboot 目录。

步骤 3 配置加密 KEY 和 IV（可选）。

如果要生成加密安全启动 uboot 镜像，进入 source/bootloader/secureboot 目录，设置 create_secure_boot_v3.sh 或 create_secure_boot_v4.sh 文件中的 ROOT_KEY0 和 BOOT_IV 的值。脚本中预置了加密 KEY 和 IV 值，用户可自行修改。

注：XM7606V13 使用签名脚本 create_secure_boot_v3.sh。

XM7206V11A 使用签名脚本 create_secure_boot_v4.sh。

步骤 4 生成安全 boot 镜像。

```
cd source/bootloader/secureboot
```

➤ **生成普通安全 boot 镜像：**

```
make CHIP=xxx AES_ENC=false
```

➤ **生成加密安全 boot 镜像：**

```
make CHIP=xxx AES_ENC=true
```

最终在 source/bootloader/secureboot/output 目录下生成对应的安全镜像。

----结束

注意：

发布包脚本会在第一次编译时产生公钥和私钥文件，后续编译的安全镜像均采用第一次生成的公钥和私钥，如果要更新公钥和私钥，需手动删除 rsa4096pem 目录下的文件。

2.2 密钥文件介绍

```
└─ rsa4096pem
|   └─ rsa4096_pem_hash_val.txt  //文本格式的公钥 HASH 和寄存器配置命令
|   └─ rsa_priv_4096.pem        //PEM 格式私钥
|   └─ rsa_pub_4096.bin         //二进制格式公钥
|   └─ rsa_pub_4096.pem        //PEM 格式公钥
|   └─ rsa_pub_4096_sha256.txt  //文本格式的公钥的 HASH 值
└─ aes_otp_cfg.txt             //AES KEY 寄存器配置命令
```

3 OTP 烧写介绍

3.1 操作步骤

在安全启动方案中，需要烧写三部分数据到 OTP 区域，包括：

- RSA 公钥的哈希值（必选）
- 安全启动控制位（必选）
- 加密用 AES KEY（可选）

XM7606V13 OTP 区域烧写

整体烧写步骤如下：

步骤 1 烧写非安全 U-boot，并启动 U-boot 至命令行

步骤 2 RSA 公钥哈希值烧写（必选）

```
mw 0x10090400 0x6
mw 0x1009040c 0XXXXXXXXX
mw 0x10090410 0XXXXXXXXX
mw 0x10090414 0XXXXXXXXX
mw 0x10090418 0XXXXXXXXX
mw 0x1009041C 0XXXXXXXXX
mw 0x10090420 0XXXXXXXXX
mw 0x10090424 0XXXXXXXXX
mw 0x10090428 0XXXXXXXXX
```

注意：以上公钥 HASH 配置命令，可从 rsa4096_pem_hash_val.txt 中直接 copy

```
mw 0x10090408 0x2
```

mw 0x10090404 0x1adde888

步骤 3 加密用 AES KEY 烧写 (可选)

mw 0x10090400 0x0

mw 0x1009040c 0xFFFFFFFF

mw 0x10090410 0xFFFFFFFF

mw 0x10090414 0xFFFFFFFF

mw 0x10090418 0xFFFFFFFF

mw 0x1009041c 0xFFFFFFFF

mw 0x10090420 0xFFFFFFFF

mw 0x10090424 0xFFFFFFFF

mw 0x10090428 0xFFFFFFFF

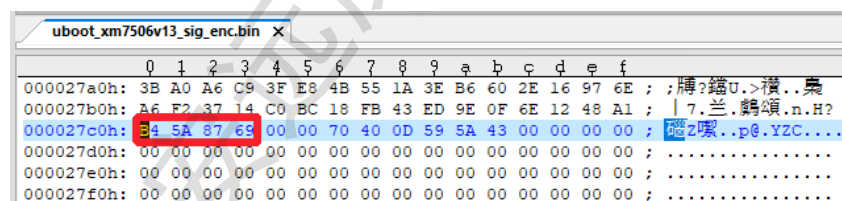
注意：以上 AES KEY 配置命令，可从"aes_otp_cfg.txt" 中直接 copy。烧写普通安全 uboot 镜像时不需要执行本步骤，烧写加密安全 uboot 镜像时才需要执行本步骤。

mw 0x10090408 0x2

mw 0x10090404 0x1adde888

步骤 4 安全启动模拟验证 (可选)

修改签名后的 uboot 0x27c0 处的值，修改 0x69875AB4 为其它值，然后将修改后的 uboot 镜像烧写至启动介质，上电后查看是否能正常启动。若能正常启动，可执行步骤 5。若启动失败，请勿执行步骤 5，否则单板将成“板砖”。



	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
000027a0h:	3B	A0	A6	C9	3F	E8	4B	55	1A	3E	B6	60	2E	16	97	6E ; ; 牌? 籍U.> 横.. 鼻
000027b0h:	A6	F2	37	14	C0	BC	18	FB	43	ED	9E	0F	6E	12	48	A1 ; 7. 兰. 麟. 领.n.H?
000027c0h:	5A	87	69	00	00	70	40	0D	59	5A	43	00	00	00	00	00 ; 磁z 嘿...p@.YZC....
000027d0h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00 ;
000027e0h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00 ;
000027f0h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00 ;

步骤 5 安全启动控制位烧写 (必选)

mw 0x10090430 0x0

mw 0x10090434 0x1

mw 0x10090408 0x4

mw 0x10090404 0x1adde888

步骤 6 安全镜像烧写

通过 U-boot 命令行烧写安全镜像至启动介质，或通过烧写工具烧写安全镜像至启动介质，完成安全镜像的烧写。

----结束

XM7206V11A OTP 区域烧写

整体烧写步骤如下：

步骤 1 烧写非安全 U-boot，并启动 U-boot 至命令行

步骤 2 RSA 公钥哈希值烧写（必选）

```
mw 0x100a0008 0x4
mw 0x100a000c 0XXXXXXXXX
mw 0x100a0010 0XXXXXXXXX
mw 0x100a0014 0XXXXXXXXX
mw 0x100a0018 0XXXXXXXXX
mw 0x100a001c 0XXXXXXXXX
mw 0x100a0020 0XXXXXXXXX
mw 0x100a0024 0XXXXXXXXX
mw 0x100a0028 0XXXXXXXXX
```

注意：以上公钥 HASH 配置命令，可从 rsa4096_pem_hash_val.txt 中直接 copy

```
mw 0x100a0000 0x3
mw 0x100a0004 0x1acce551
```

步骤 3 加密用 AES KEY 烧写（可选）

```
mw 0x100a0008 0x0
mw 0x100a000c 0XXXXXXXXX
mw 0x100a0010 0XXXXXXXXX
mw 0x100a0014 0XXXXXXXXX
mw 0x100a0018 0XXXXXXXXX
mw 0x100a001c 0XXXXXXXXX
mw 0x100a0020 0XXXXXXXXX
mw 0x100a0024 0XXXXXXXXX
mw 0x100a0028 0XXXXXXXXX
```

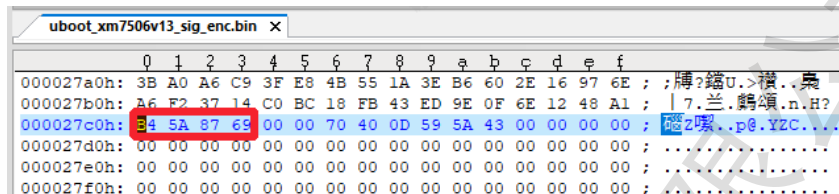
注意：以上 AES KEY 配置命令，可从 "aes_otp_cfg.txt" 中直接 copy。烧写普通安全 uboot 镜像时不需要执行本步骤，烧写加密安全 uboot 镜像时才需要执行本步骤。

```
mw 0x100a0000 0x3
```


mw 0x100a0004 0x1acce551

步骤 4 安全启动模拟验证（可选）

修改签名后的 uboot 0x27c0 处的值，修改 0x69875AB4 为其它值，然后将修改后的 uboot 镜像烧写至启动介质，上电后查看是否能正常启动。若能正常启动，可执行步骤 5。若启动失败，请勿执行步骤 5，否则单板将成“板砖”。



步骤 5 安全启动控制位烧写（必选）

mw 0x100a0034 0x0

mw 0x100a0030 0x1

mw 0x100a0000 0x5

mw 0x100a0004 0x1acce551

步骤 6 安全镜像烧写

通过 U-boot 命令行烧写安全镜像至启动介质，或通过烧写工具烧写安全镜像至启动介质，完成安全镜像的烧写

----结束

务必注意：

- 以上每个烧写步骤都必须谨慎小心，以免烧写错误导致芯片不可用。
- 十六进制的地址“0xXXXXXXXX”中，前缀 0x 中的 x 必须是小写字母。