原创 已于 2025-12-08 15:47:42 修改 · 1.3k 阅读 · 23 · 17 · CC 4.0 BY-SA版权

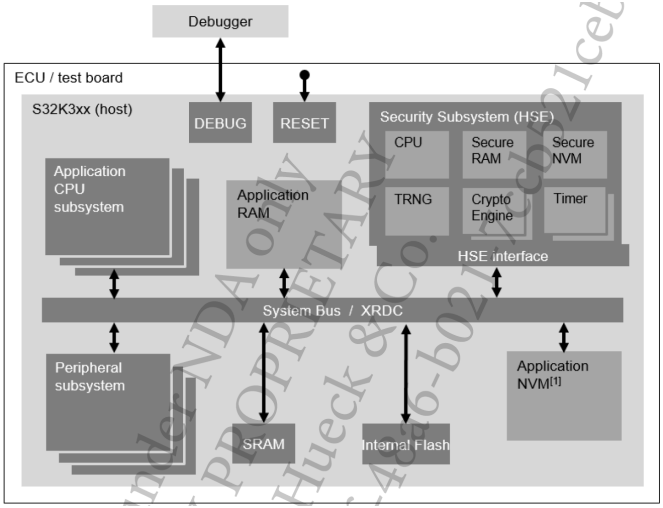
文章标签： #网络安全

目录

1. 系统架构

NXP S32K3xx系列芯片可以分为：

- 应用域：即主机Host部分
- 安全域：即HSE，提供硬件安全引擎



从上图可以看出，应用域包含以下的系统资源：

- 若干外设资源如通信接口，定时器，编码/解码器
- 与片外内存的接口
- 系统总线，所有系统资源相互连接

安全域即HSE系统，具有专属的系统资源，通过专用接口连接到主机部分，具体请参考1.2章节。

1.1 主机Host

1.1.1 CPU 子系统

主机的CPU子系统包含一个CPU核和专用的CPU资源(cache，中断控制器、浮点单元等)。它可以处理如可执行文件，配置数据和应用数据等内容过控制系统资源来实现ECU的功能。

主机可以包含多个CPU子系统，即多核系统。

1.1.2 内存资源

1. Application RAM: CPU子系统可以访问的片上和外部RAM 区域

2. Application NVM: 包括以下几种

- One-time-programmable内存: 可参考S32K3xx Reference Manual手册的附件**DCF配置**
- 片上的Code和Data Flash
- Device memory configuration：配置UTEST相关参数，如下表所示：

配置参数	UTEST地址	大小	说明
HSE Firmware Usage feature flag	0x1B000000	8 bytes	使能安装HSE固件
Reserved	0x1B000048	8 bytes	预留
FXOSC configuration	0x1B000050	8 bytes	在安全启动阶段使能PLL
Partial AB_Swap	0x1B000058	8 bytes	使能部分AB SWAP配置，支持的芯片为S32K358和S32K388
JDC clock disable	0x1B000060	8 bytes	禁止JDC时钟，降低功耗

1.1.3 设备标识符UID

NXP提供64bit的唯一设备标识符，通过UID可以识别设备。详细信息可参考S32K3xx Reference Manual手册的附件**DCF配置**

1.1.4 主机系统镜像

嵌入式Flash中包括以下的主机系统镜像：

- IVT: Image Vector Table，是复位后系统执行的入口
- Apps：应用程序镜像，如可执行文件，数据等
- AppBL: 经过认证的应用程序镜像，在HSE认证成功后才可以运行

1.1.5 系统总线和XRDC

系统总线是CPU子系统与系统资源如RAM和内部Flash通信的桥梁，而XRDC(eXtended Resource Domain Controller)可以控制CPU子系统**访问权限**。应用程序可以通过XRDC配置去定义CPU子系统对系统资源的访问权限。

1.2 HSE子系统

HSE是一个安全的子系统，主要目标是为具有严格的保密性和真实性要求的应用程序提供相关的安全功能，如以下的应用场景：

- 为应用程序(主机)保存安全敏感的信息（例如，密钥值）
- 通过专用处理器来进行加密等操作
- 在运行时和系统启动期间实施安全措施

1.2.1 HSE子系统类型

- HSE_H (High)：支持的芯片系列为S32G2, S32G3, S32ZSE和S32R45
- HSE_M (Medium)：支持的芯片系列为S32R41和SAF85
- **HSE_B (Base)**：支持的芯片系列为S32K3XX，本文介绍的即为HSE_B类型

1.2.2 HSE子系统特性

1. CPU子系统

HSE的CPU子系统通过调用系统资源来为主机提供安全的服务，服务的API信息，可参考<**HSE Service API Reference Manual**>

2. 加密加速器

HSE子系统提供了以下的加密算法的加速器：

- AES引擎：支持所有key的大小（128，192，256bits)和多种复杂的模式（CBC,CTR,GCM等）
- Hash引擎：支持标准SHA1和SHA2哈希，最高可达256位摘要的哈希。对于SHA-384和SHA-512，软件支持可用
- 公钥引擎：加速RSA和ECC操作

3. 真随机数生成器

HSE子系统支持真随机数发生器(TRNG)

4. 系统定时器

HSE子系统配备系统定时器，允许自动运行功能，如运行时内存验证检查，以及一个看门狗定时器，在**发生意外运行时故障时重置HSE子系统**

1.2.3 HSE内存资源

- Secure RAM: 由HSE子系统独占访问的RAM资源，用来运行hse和存储拷贝的加密密钥。
- Secure NVM: 由HSE子系统独占访问的NVM资源，包括如代码，数据和配置的flash区域

1.2.4 HSE的镜像

HSE的镜像包括：

- FW-IMG：HSE固件的可执行文件，存储在HSE的code flash区域
- SYS-IMG：包含公/私钥，monotonic计数器和配置数据(即HSE系统属性)，存储在HSE的data flash区域

1.2.5 生命周期

生命周期是内部设备的状态，由HSE子系统进行管理，LC的状态可以读取，也可以通过HSE的系统属性服务进行修改（注意只能向下一阶段演进退），LC也可以通过IVT的LCW进行演进。

不同LC状态的描述如下表所示：

Table 4. LC states

LC state	Description
CUST_DEL	Device (that is, NXP IC) delivered to system integrator (that is, NXP's customer) for ECU manufacturing and initial configuration.
OEM_PROD	ECU (device) delivered to the OEM for vehicle integration and final configuration.
IN_FIELD	ECU integrated in the vehicle and operating; it is the state of normal device use (and most secure state).
PRE_FA	Similar to IN_FIELD; Provides capabilities for failure analysis.
FA	ECU (device) failure; this is the state for functional testing of the IC.

以上LC的状态为**FA lifecycle和PRE_FA life cycle只能由NXP来演进。**

1.2.6 软件组件

HSE子系统包含如下两个软件组件：

1. SBAF

Secure Boot Assist Flash (即SBAF)是在生产阶段由NXP写入的，该软件组件存储在HSE的code flash区域，主要包括如下特性：

- 安全启动模式和非安全启动模式
- 应用程序启动核心选择
- HSE固件安装

关注

- 设备OTA功能启用
- 芯片生命周期演进
- 调试Debug认证
- 部分分区切换启用
- XRDC配置
- 支持固件更新
- 安全和基于JTAG的恢复模式
- HSE固件握手
- 支持HSE固件更新
- ECC错误检测/处理

2. HSE固件

HSE固件主要提供以下服务：

- Administration Service：用于安装、配置和测试HSE
- Key management Service: 用于管理不同的密钥
- Cryptographic Service: 为应用提供密码学原语，用于实现应用层的加密协议栈
- Random Number Service: 生成可用于各种安全协议的随机数
- Memory verification Service: 允许应用程序在启动时(复位后)和运行时验证不同的内存区域
- Monotonic counter Service: 为应用程序提供一组可读取的单调计数器，且只能逐渐增加

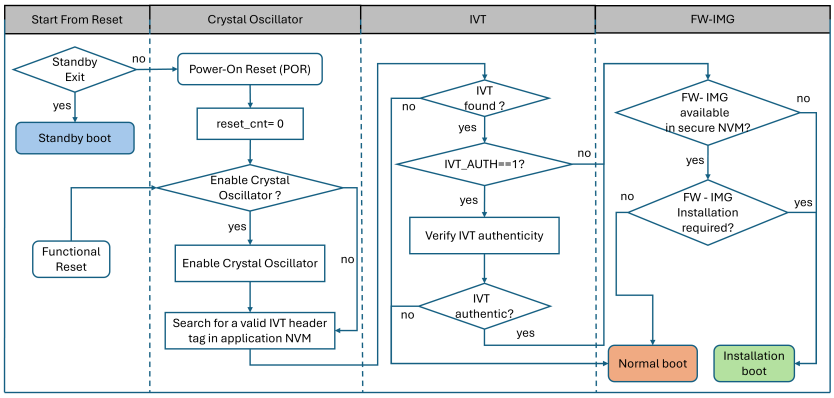
HSE 固件在出厂后没有激活，需要用户进行HSE 固件安装，HSE安装方法请参考NXP S32K3xx之HSE使用方法（一）

2. 启动流程

启动阶段，HSE系统最终会运行normal boot或者Installation boot，取决于以下内容：

- IVT是否存在
- FW-IMG是否存在
- 是否需要安全FW-IMG

从系统复位开始，这里暂时不考虑standby模式，如果为Power on reset，对复位次数进行计数清0，功能复位则跳过该步骤。然后使能晶振，判
有效以及是否需要进行IVT认证（IVT_AUTH），如果IVT无效且认证失败，进入Normal boot；否则判断在Secure NVM中是否存在有效的FW-IMG，如
的FW-IMG或者存在FW-IMG安装请求，则进入Installation boot。



2.1 IVT介绍

IVT全称为Image Vector Table ，也可称为boot header，是复位后系统主要的执行入口，包含Apps的存储位置，BCW配置(包含启动时的行为)，(允许进行生命周期的演进)，接下来对IVT进行详细的介绍。

2.1.1 IVT start Address

关注

Device	Start addresses (FULL_MEM)	Start addresses (AB_SWAP)	Start addresses (Partial AB_SWAP)
S32K310	0x00400000, 0x10000000	0x00400000, 0x10000000	NA
S32K311	0x00400000, 0x00480000, 0x10000000	0x00400000, 0x10000000	NA
S32K341	0x00400000, 0x10000000	0x00400000, 0x10000000	NA
S32K312 S32K342 S32K322	0x00400000, 0x00500000, 0x10000000	0x00400000, 0x10000000	NA
S32K344 S32K324 S32K314	0x00400000, 0x00500000, 0x00600000, 0x00700000, 0x10000000	0x00400000, 0x00500000, 0x10000000	NA

2.1.2 IVT Structure

IVT的结构如下表所示，需要注意的是：

- Apps的地址指针需与VTOR相匹配
- 所有其他的地址指针必须32字节边界对齐
- 预留的区域填充0xFF

Offset	Byte size	Category	Description	Value/Value type
0x00	4	Tag	IVT header tag	Magic number(0x5AA55AA5)
0x04	4	Configuration	**BCW**	Bit field
0x08	4	Reserved		
0x0C	4	Executable	Apps for BOOT_TARGET bit #0	Pointer
0x10	4	Reserved		
0x14	4	Executable	Apps for BOOT_TARGET bit #1	Pointer
0x18	4	Reserved		
0x1C	4	Executable	Apps for BOOT_TARGET bit #2	Pointer
0x20	4	Reserved		
0x24	4	Configuration	**LCW**	Pointer
0x28	4	Executable	Apps for BOOT_TARGET bit #8	Pointer
0x2C	4	Executable	FW_IMG	Pointeronly, only valid for FULL_MEM configuration
0x30	4	Executable	**AppBL**	Pointer
0x34	12	Reserved		
0x40	4	Executable	Start Address of Application Core for Secure Recovery mode.	Pointer
0x44	4	Length	Length of Recovery Application	32-bits data
0x48	156	Reserved		

0xE4	12	vector (IV)	GMAC.IV value is also included in GMAC calculation.	randomly generated every time the GMAC is calculated
0xF0	16	TAG	Authentication tag (GMAC)	Byte array

如果IVT未提供，可以将FW-IMG放在0x00400000开始进行HSE的固件安装。

1. Boot Configuration Word (BCW)

BCW的详细定义请参考**REF02**文档，现只针对常用的配置进行说明：

- BOOT_TARGET：使能应用核
- BOOT_SEQ：安全启动是否使能
- PLL_ENABLE：用于在安全启动期间使能PLL，只有BOOT_SEQ == 1时配置
- SWT0_ENABLE：用于在应用核ungating之前启用Application SWT0

2. Lifecycle Configuration Word (LCW)

LCW为32bit值，定义了LC演进状态：

- LCW = 0xDADADADA将LC演进至OEM_PROD
- LCW = 0xBABABABA将LC演进至IN_FIELD

当没有安装HSE固件时，可以使用LCW进行生命周期的演进，如果安装了HSE，可以使用HSE的系统属性管理服务进行生命周期状态的管理。

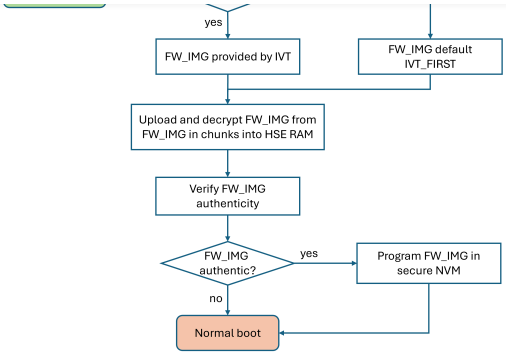
3. AppBL Structure

Offset	Byte size	Category	Description	Value / Value type
0x00	1	Tag	AppBL header tag	Magic number (0xD5)
0x01	2	Reserved		
0x03	1	Tag	AppBL version	Magic number (0x60)
0x04	4	Reserved		
0x08	4	Configuration	Start address (in Flash)	Pointer
0x0C	4	Configuration	AppBL size (N)	32-bit integer
0x10	1	Configuration	Core identifier	Value
0x11	47	Reserved		
0x40	N	Executable	AppBL content (in plain)	Executable
N +0x40	12	IV	12-byte random vector	Byte array
N +0x4C	16	Tag	Authentication tag (GMAC)	Byte array

2.2 Installation Boot

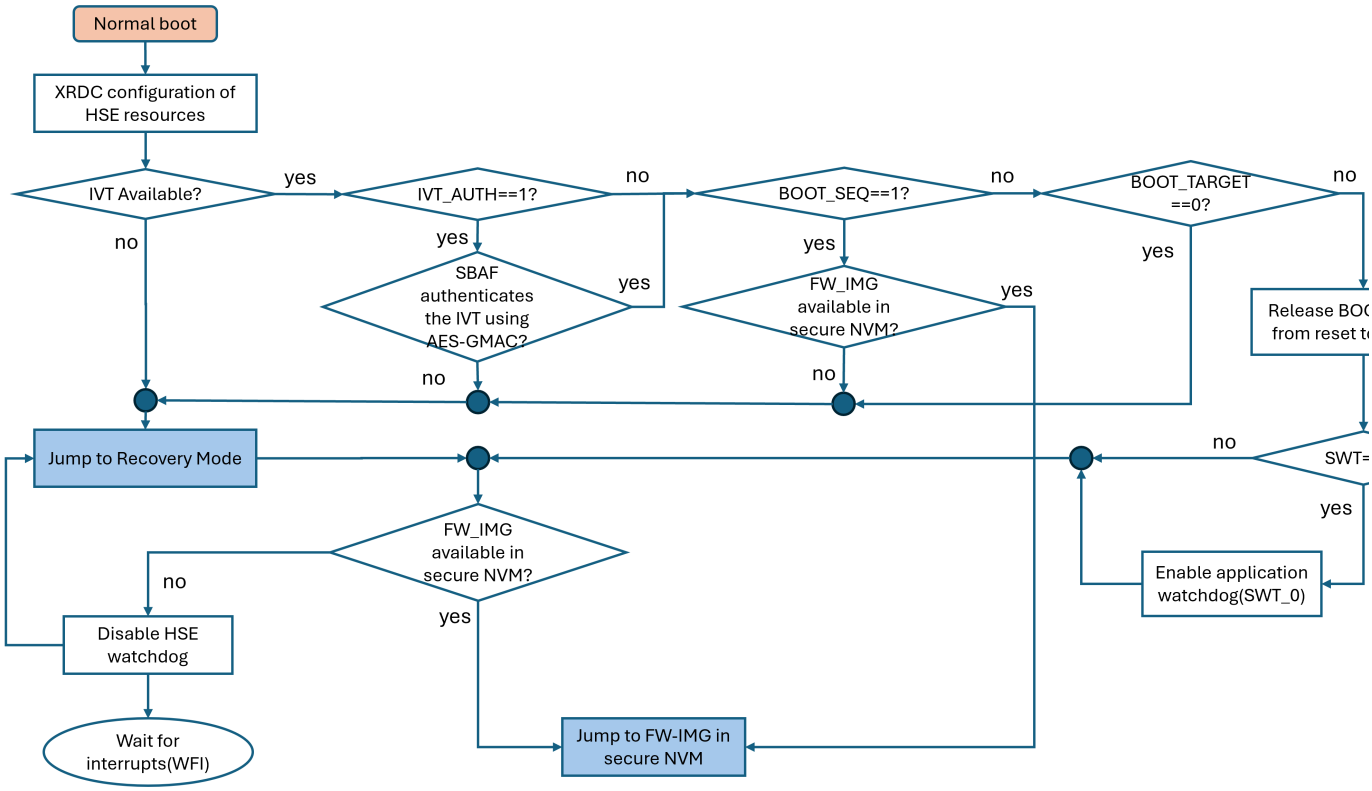
在Installation boot阶段，HSE从Applicaiton NVM中读取加密的FW-IMG，进行解密并且存储在受保护的系统RAM中。如果该解密的文件经过了真证，HSE系统将其写入code flash中，同时将RAM中的内容删除。

FW-IMG的地址通过IVT提供(参考2.1节的IVT structure)，或者将FW-IMG放置在IVT的起始位置即IVT_START中。



2.3 Normal Boot

在Normal boot流程中，HSE系统进行IVT的加载和解析，根据BOOT_SEQ和BOOT_ TARGET配置参数进行处理，具体流程如下图所示：



如果secure NVM中存在HSE固件，那么最终会运行HSE，其他情况则进入wait-for-interrupt模式（WFI）

以下异常情况会导致主核进入Recovery Mode，详细内容请参考文档**REF02**

- IVT不存在或者被损坏
- 连续超过8次的复位(functional or destructive复位)
- 应用软件的安全启动认证失败

3. 安全启动

3.1 安全启动类型

1. Basic Secure Boot（IVT based secure boot)

- 只支持一个目标核(AppBL)
- 只有SMR/CR不可用时才可以使用(没有配置CR)
- 所有的参数定义在IVT中，如入口地址和大小，在AppBL的header中定义，不需要更行HSE SYS_IMG
- AppBL镜像可以使用GMAC进行认证，使用ADKP派生的key。GMAC的tag使用相关服务生成或者使用离线工具，如果使用离线工具，random IV须添加在AppBL的最后。

- 支持单核或多核（定义在Core Reset Table）
- 基于SMR(Secure Memory Region) 和CR (Core Reset) 配置
- 支持对称加密认证（AES-CMAC, GMAC, HMAC等）
- 支持RSA，ECDSA和EDDSA签名验证

3. SHE Based Secure Boot

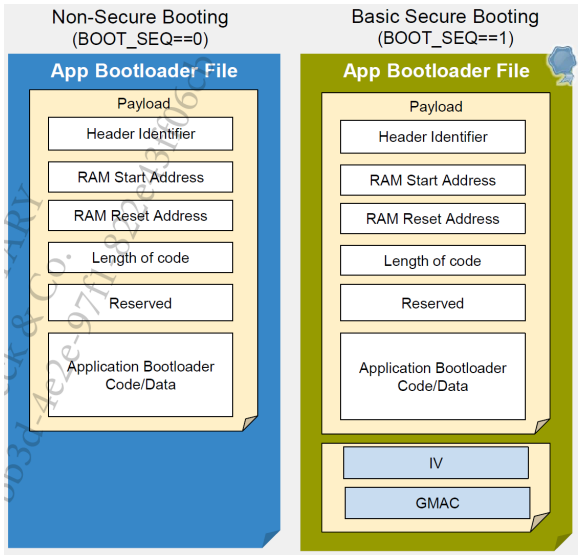
- 模拟基于SHE协议的安全启动
- 只有一个应用程序的核在复位后执行
- SHE仅支持基于CMAC的认证方案，使用BOOT_MAC_KEY的密钥
- 它是SMR安全启动的变种方案，使用SMR的第一个入口(索引为0)。HSE固件通过读取SMR#0的key handle来识别是SHE安全启动。如果SMR#0 handle是SHE BOOT_MAC_KEY，HSE固件会启动SHE安全启动。

如果**SMR与Basic Secure Boot（已配置AppBL）一起配置**，并且在启动时，SYS_IMG加载失败（SMR不可用）或没有CR入口表时，应用核心仍可以使用AppBL的镜像，**所以AppBL镜像可以看作恢复镜像。**

如果需要验证IVT，那么需设置IVT_AUTH属性，通过HSE_ENABLE_BOOT_AUTH_ATTR_ID，认证的tag使用离线工具或者使用服务生成

3.2 Basic安全启动

Application Bootloader有两种方式加载，一种是通过**BootROM(Non-Secure boot)**或**HSE FW（Secure boot)**。App Bootloader可以通过HSE的hseBootDataImageSignSrv_t的服务生成GMAC值，将生成的**随机的IV**和**tag**值存储在App Bootloader文件的最后，如下图所示：



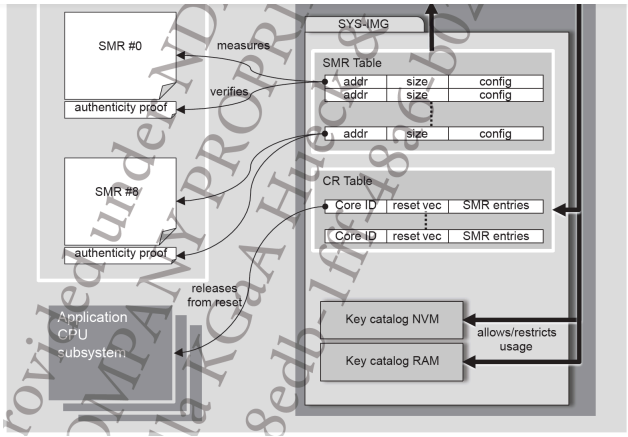
注意: 如果配置了CR入口表，那么忽略Basic Secure Boot 启动。

3.3 基于SMR安全启动

Secure memory region(SMR)定义了host内存中需要进行认证的安全内存区域，如下图中的SMR#0等，在SMR Table表中存储了SMR#n的地址和size以相关的配置config。图中authenticity proof是SMR#n区域内存数据的认证值，如MAC值或RSA/ECC签名。

Core Table是执行SMR#n区域验证后需要运行的核和复位向量表，为复位之后运行的入口地址，具体细节参考3.3.2章节。

在SMR#n区域进行认证的过程中，需要使用key catalog NVM和key catalog RAM中的key，关于key的介绍请参考NXP S32K3xx之HSE使用方法 CSDN博客

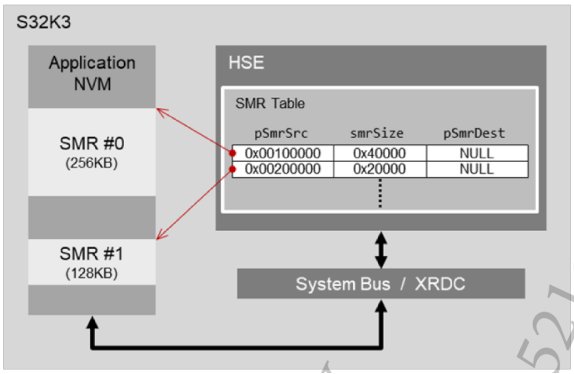


3.3.1 SMR Table

SMR表中定义了若干属性，每个属性的定义和描述如下表所示：

Attribute	Data field	Description
Source address	pSmrSrc	SMR Source Address (External or internal Flash)
Size	smrSize	SMR Size
Destination address	pSmrDest	SMR destination address (System RAM)
Initial authenticity proof	pInstAuthTag[]	Initial Auth. Tag address (External or internal Flash)
Authentication scheme	authScheme	Verification Scheme (MAC, RSA, ECC)
Authentication key	authKeyHandle	NVM Key
Decryption parameters	smrDecrypt	Reference to SMR decryption values
Verification period	checkPeriod	Define the verification sequence period
SMR configuration flags	configFlags	Configuration flags for memory interface and the authenticity proof
SMR Version	versionOffset	offset in SMR where the image version can be found

以如下图为例，SMR表的第一行即为SMR#0的源地址pSmrSrc: 0x00100000，大小smrSize为0x40000(256KB)，第二行为SMR#1的源地址pSm 0x00200000，大小smrSize为0x20000(128KB)。



Host可以通过请求SMR安装的服务进行安装，定义**hseSmrEntryInstallSrv_t**结构体，主要包括以下内容，详细的安装过程请参考文档**REF02**

- 定义SMR属性和SMR的入口entry
- 提供SMR内容的认证proof
- SMR内容加密后的认证proof(可选)

3.3.2 CR Table

CR表中定义了若干属性，每个属性的定义和描述如下表所示：

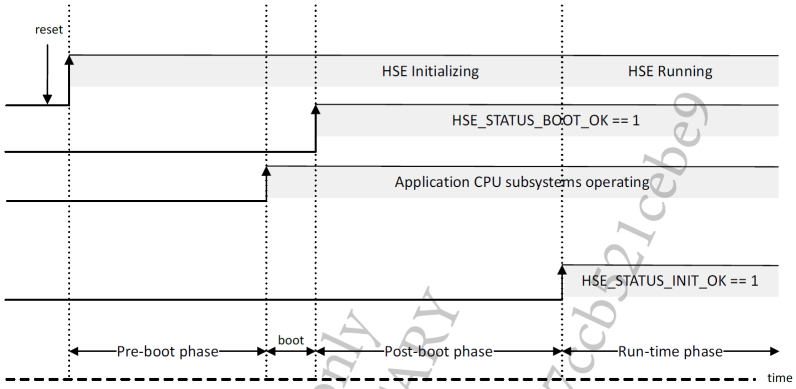
Pre-boot SMR verification map	preBootSmrMap	A set of flags that define which SMR, indexed from 0 to 31(bit #i for SMR #i)
Alternate Pre-boot SMR verification map	altPreBootSmrMap	A set of flags that define which SMR, indexed from 0 to 31 (bit #i for SMR #i)
Post-boot SMR verification map	postBootSmrMap	A set of flags that define which SMR, indexed from 0 to 31 (bit #i for SMR #i)
Reset address	pPassReset	A Value of the VTOR of associated application subsystem
Alternate reset address	pAltReset	Value of the VTOR of associated application subsystem if all the SMR defined in altSmrVerifMap pass the verification
Core boot option	startOption	Specifies whether the core is automatically started by the HSE at boot-time or if the CR entry is used for on-demand booting at run-time.
Sanctions on failed verification	crSanction	The sanction HSE applies for the CR entry if one of the associated SMR fails verification.

以上CR的属性中包含Pre-boot，Alternate Pre-boot和Post-boot的概念，在下一小节介绍。

Host可以通过请求CR安装的服务进行安装，定义**hseCrEntryInstallSrv_t**结构体，详细的安装过程请参考文档**REF02**

3.3.3 Secure Boot phase

安全启动包含以下几个阶段，如下图所示：



1. Pre-boot phase

在Pre-boot阶段，HSE从CR table的第一个入口索引开始进行解析。对于每个一个CR的入口，通过*pCrEntry->PreBootSmrMap*链接的SMR区域
果任意SMR验证失败，那么HSE开始验证通过pCrEntry->altPreBootSmrMap链接的SMR区域。

如果pre-boot阶段的SMR区域验证成功，HSE开始启动CPU子系统，具体的启动策略参考*core reset release strategies*

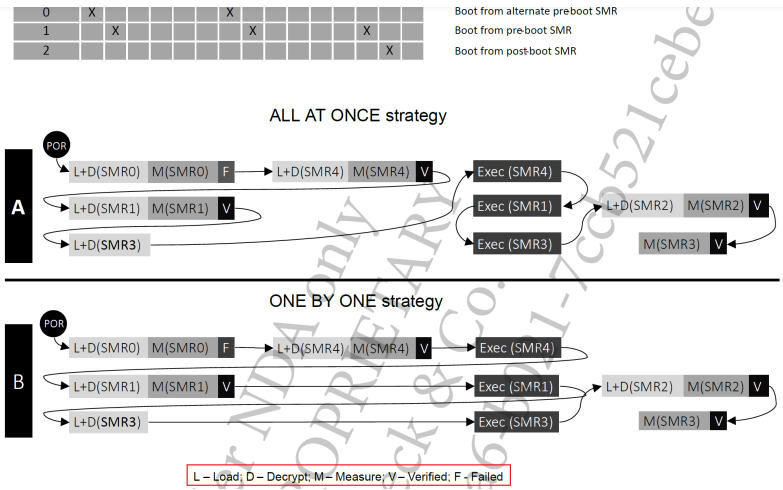
如果至少有一个pre-boot阶段的SMR区域验证失败，HSE使用CR入口的sanction进行处理。

2. Booting phase and core reset release strategies

根据hseAttrCoreResetRelease_t属性，有以下两种处理方式：

- ALL_AT_ONCE：首先HSE解析CR表的所有入口，验证所有相关的pre-boot SMR区域，然后启动所有通过验证的CPU子系统
- ONE_BY_ONE：在相关的CR表入口核pre-boot SMR验证成功后，HSE逐个启动CPU子系统

两种方式的启动流程对比如下图所示：



pre-boot阶段在第一个CPU子系统启动时结束。

Pre-boot和Boot阶段结束时，所有已配置的CPU子系统都被启动，HSE通过状态标志HSE_STATUS_BOOT_OK来通知。

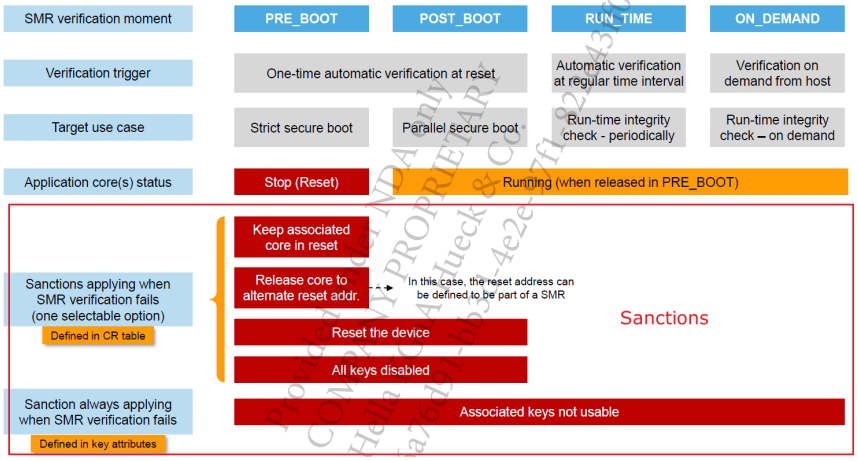
3. Post-boot phase

在所有配置的CPU子系统已经启动且boot阶段之后，HSE通过CR表的pCrEntry->postBootSmrMap，对链接的SMR区域进行验证。如果验证失败
置的sanction进行处理。

4. Sanctions

SMR验证失败后，有以下两种惩罚方式：

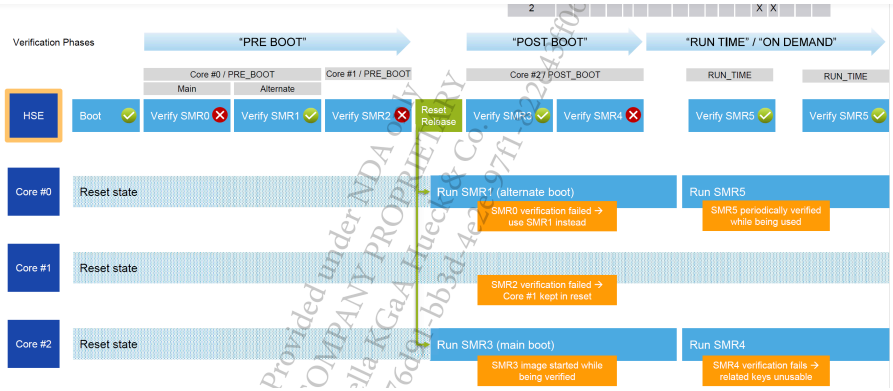
- Key使用惩罚
- 设备操作惩罚



关于sanctions的详细内容请参考文档**REF02**

5. Example

以下是安全启动的流程图，包含各个验证阶段的HSE和CPU子系统的状态和操作。



参考文档

REF01: <an781201-Secure Boot Overview Training(0.1).pdf>

REF02: <RM758226-RM00286 HSE-B Firmware Reference Manual - V2.6(2.6).pdf>

S32K324 HSE固件加载配置及说明 赞的
本文介绍HSE固件的下载，及利用S32DS工程及PE工具安装。（本文只讨论对于full_men方式的HSE）HSE作为信息安全实现的载体，还是有很多需要学习的地方。

汽车信息安全--S32K3的HSE如何与App Core通信(1)? 认真搞搞汽车N
本章内容讲述了S32K Host和Hse之间的通信机制，就原理来看，更像是TC3xx HSM、RH850 ICU中HT2HSM、HSM2HT的升级版本。 这也给我们在做这方面的软硬件设计

汽车信息安全 -- 再谈车规MCU的安全启动_mcu安全启动
1.3NXPS32K3的安全启动 在S32K3的启动流程,硬件复位后同样只有HSE(也就是HSM)子系统可以运行,首先运行sBAF()代码,完成基本环境配置,然后根据IVT(Image Vector t

S32K3芯片HSE功能验证之安装HSE FW_s32k3 hse
HSE(Hardware Security Engine)是NXP S32K3系列微控制器中的一个安全模块,提供了一系列安全特性,如加密、安全启动和密钥存储。 主要功能加密:支持多种加密算法,包括

S32K3的示例例程与hse等
S32K3的示例例程与hse等

S32K3系列安装HSE的例程代码通过IVT的方式实现
S32K3系列安装HSE的例程代码通过IVT的方式实现,生成的HEX脱机运行断电上电两次就行，要是实现代码直接编译，安装一下RTD的1.0.0版本的SDK。

S32K3 學習筆記_s32k3 ivt
IVT 解析 Boot Configuration Word 配置 RTD工程啟動所需文件 RTD啟動流程圖 總結 2.S32K3 FOTA Use cases HSE FW Update Procedure for Full Mem HSE FW Update Pr

NXP应用随记(七):S32K3XX复位与启动阅读记录_sbaf
sBAF,或称为安全引导辅助固件(Secure Boot Assist Firmware),是NXP S32K3微控制器系列中的一个功能。它是一种固件,用于在微控制器启动时提供安全功能,如验证应用程

从车灯模组的角度聊聊信息安全需求 北极熊的脖子（16605192620@163.co
最近在和一些车灯客户交流时，发现很多车灯项目都多了信息安全的需求，为了进一步了解信息安全的需求，笔者收集了信息安全相关的文档进行学习和梳理。下文是笔者

NXP S32K3xx之HSE使用方法（一） **最新发布** weixin_46848690的
本文针对NXP的S32K3系列的HSE固件安装及使用HSE进行信息安全开发进行了介绍，部分内容参考NXP官网提供的资料

S32K3安全启动实现[源码]_s32k3hse资源
本文详细介绍了S32K3安全启动的三种配置机制:基本安全启动(Basic Secure Boot)、高级安全启动(Advanced Secure Boot)和基于SHE的安全启动(SHE based Secure Boot

在IAR Embedded Workbench for Arm中实现NXP S32K3安全调试
随着汽车电子系统变得越来越智能,对功能安全(Safety)的要求越来越高,同时信息安全(Security)也越来越被关注,安全调试(Secure Debug)机制已成为一个重要的信息安全特

NXP应用随记（八）：S32K3XX的HSE学习记录（HSE\MU\UTEST\IVT\A,B SWAP） 梦想技
HSE，即硬件安全引擎（Hardware Security Engine），是NXP S32K312微控制器（MCU）中的一个功能，它提供了一系列的安全特性，包括加密、安全启动和密钥存储等

HSE_ABswap双分区功能（三） AAAK_Lei的
文章以NXP的S32K344芯片为例，采用上一章提到的第二种安装方式安装，将HSE_FW安装在0x400000。

i.MX RT1170处理器系统安全 pslyunhai3255的
i.MX RT1170跨界MCU以1GHz的速度刷新了记录。该突破性系列结合了卓越的计算能力、多种媒体功能以及实时功能，易于使用。双核i.MX RT1170采用主频达1GHz的Co

小猫爪：S32K3学习笔记14-S32K3之REG_PORT,MPU和XRDC **热门推荐** Oushuwen的
小猫爪：S32K3学习笔记14-S32K3之REG_PORT,MPU和XRDC1 前言2 REG_PORT3 MPU4 XRDC 1 前言 这一节就来看看S32K3的资源访问保护器，S32K3在这一方面主

S32K3 Secure boot implement m0_54090930的
在第一次使用BOOT_MAC_KEY安装SMR #0时，如果BOOT_MAC为空（即未初始化），并 _MAC_KEY已经配置，则由HSE计算并保存在BOOT_MAC中。在BOOT_M

关注