

云途MCAL配置与使用详解之CryptoDrivers模块

作者

内容提要

1. AUTOSAR架构下的MCAL CryptoDrivers模块(Overview)

- 1.1 CryptoDrivers模块在AUTOSAR系统中的位置
- 1.2 CryptoDrivers模块的几个关键概念
- 1.3 CryptoDrivers模块的软件需求(SWS)概述

2. 云途车规MCU 相关外设介绍

- 2.1 HCU (Hardware Cryptography Unit)
- 2.2 HCU-NVR(Hardware Cryptography Unit NVR)

3. 云途MCAL CryptoDrivers模块的YCT配置

- 3.1 配置加密原语
- 3.2 配置密钥
- 3.3 配置其他

4. 云途MCAL CryptoDrivers模块的主要接口(API)

5. 云途MCAL CryptoDrivers模块的使用Tips和FAQ

- 5.1 数据对齐
- 5.2 截短输出
- 5.3 数据填充

总结

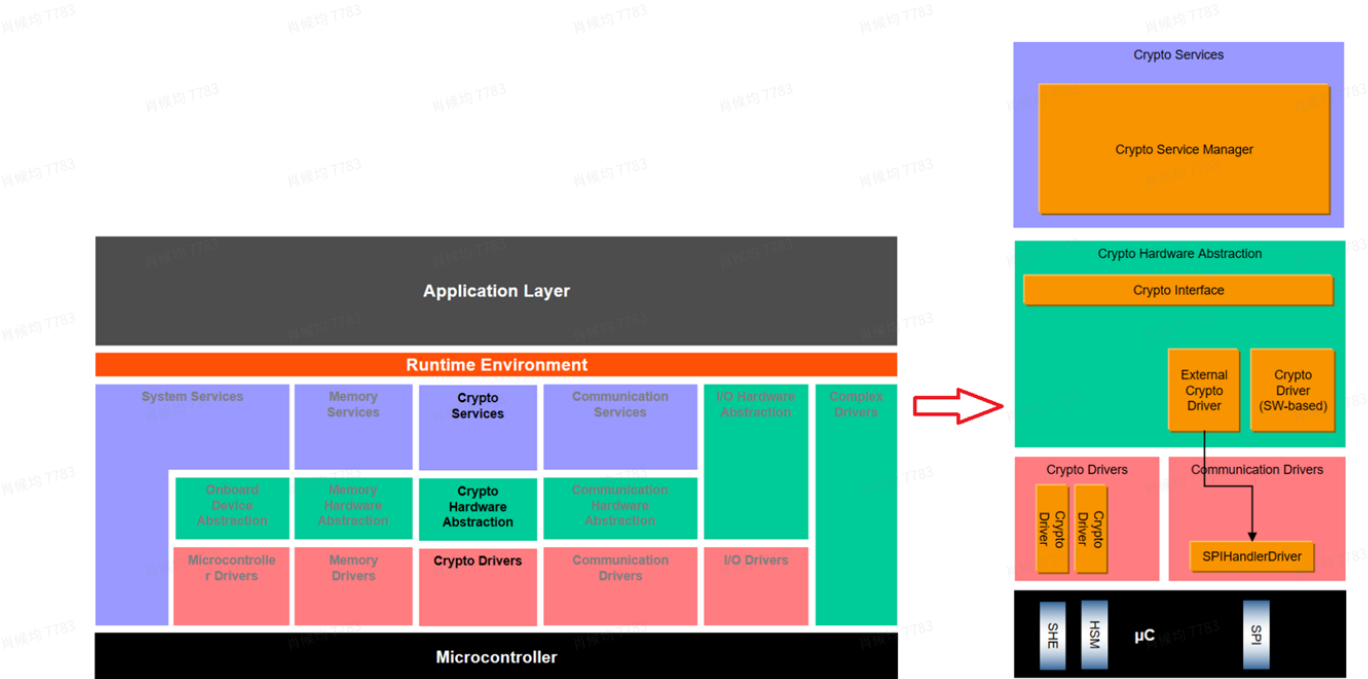
参考资料

1. AUTOSAR架构下的MCAL CryptoDrivers模块(Overview)

本章节，将介绍AUTOSAR标准文档对CryptoDrivers及关联模块的要求，以明确CryptoDrivers模块的功能和作用。

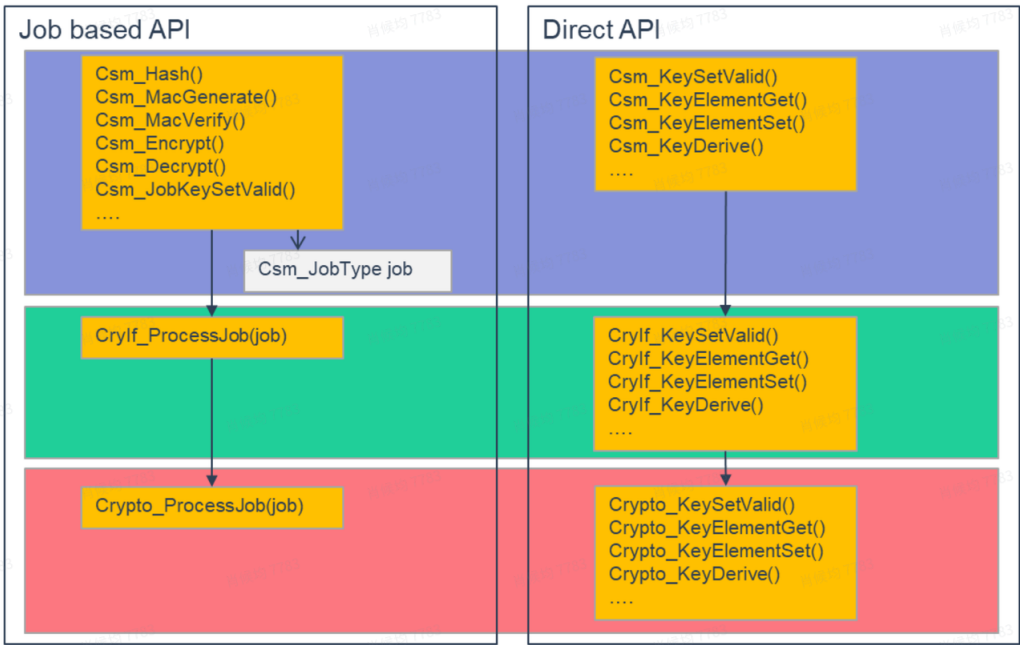
1.1 CryptoDrivers模块在AUTOSAR系统中的位置

如下图，CryptoDrivers模块属于MCAL一层，在上层模块CryIf,Csm的配合下，提供加密服务。



加密服务管理器（Csm）:Crypto Service Manager

Csm作为一个服务层，为SW-C或BSW提供接口加密操作。Csm的主要任务是管理服务，调用加密接口（CryIf）进行进一步操作。



API call tree for CSM, CryIf and Crypto. Divided into job-based API and Direct API

加密接口（CryIf）：Crypto Interface

管理1个或几个加密驱动（CryptoDriver）,无论加密驱动中的算法是硬件实现的还是软件实现的。

加密驱动：CryptoDrivers

实现同步和异步方式下的加密原语，还支持加密服务的密钥存储、配置和管理。

1.2 CryptoDrivers模块的几个关键概念

Object:加密原语的抽象。CryptoDriver由1个或多个Object组成，每一个Object实现一种加密原语，无论是硬件还是软件实现的。

KeyType: 密钥（Key）类型，由1个或多个密钥元素（KeyElement）组成。

KeyElement: 密钥元素。定义了密钥的属性及操作。

1.3 CryptoDrivers模块的软件需求(SWS)概述

从以上描述可以看出，CrptoDrivers的功能主要是两个主题：加密原语、密钥。本节从这两个主题入手，描述CryptoDrivers模块的软件需求。

加密原语

- 1、配置加密原语的算法组（AlgorithmFamily e.g. AES, MD5, RSA, ...）和算法模式（AlgorithmMode e.g. ECB, CBC, ...），配置类型是预编译（Pre-compile）。
- 2、加密原语支持同步和异步方式。
- 3、加密原语支持操作模式：“START”，” UPDATE” ，” FINISH” ，” CRYPTO_OPERATIONMODE_SINGLECALL” ，” CRYPTO_OPERATIONMODE_STREAMSTART” 。

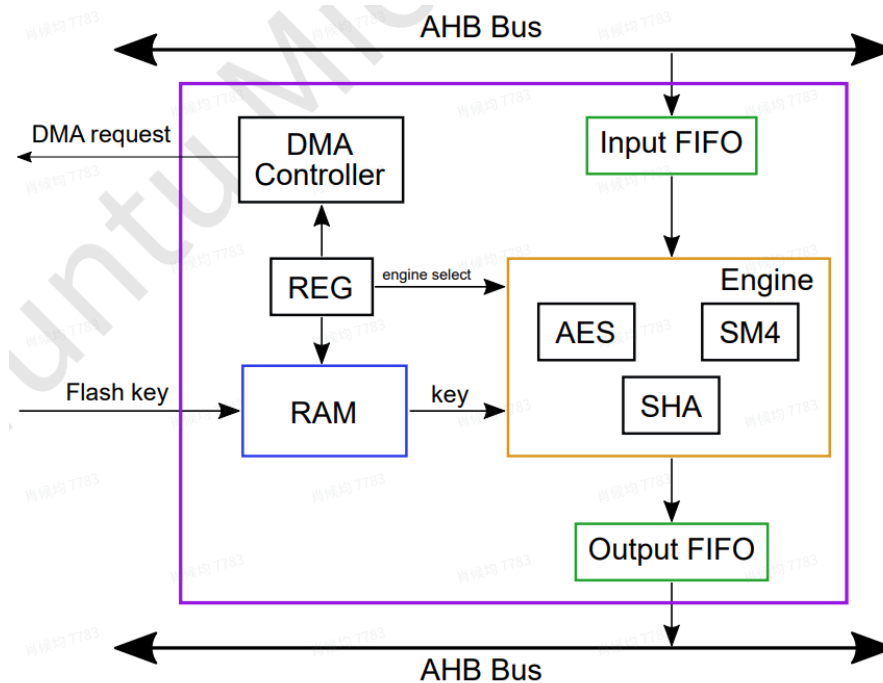
密钥

- 1、配置密钥的密钥元素（KeyElement），配置类型是预编译（Pre-compile）。
- 2、根据密钥元素（KeyElement）配置，实现密钥的存储。
- 3、以密钥元素（KeyElement）为基础，实现密钥的其他操作，例如KeyExchange，KeyCopy,KeyGenerate等。

2. 云途车规MCU 相关外设介绍

本章节，将介绍云途YTM32B1ME0车规MCU与CryptoDrivers相关硬件外设模块功能，以帮助大家更好的理解AUTOSAR CryptoDrivers模块

2.1 HCU (Hardware Cryptography Unit)



Block Diagram of HCU

支持的算法引擎：

- – AES (ECB, CBC, CTR, CCM, CMAC)
- – SM4 (ECB)
- – SHA (SHA-256, SHA-384)

主要特点：

- 支持 128-, 192- and 256-bits 密钥长度
- 支持硬件安全密钥和 灵活的软件密钥
- 支持DMA

2.2 HCU-NVR(Hardware Cryptography Unit NVR)

Table 3.1: EFM Memory Map

Memory	Start Address	End Address	Size
PFlash0	0x0000_0000	0x0007_FFFF	512 KB
PFlash1	0x0008_0000	0x000F_FFFF	512 KB
DFlash	0x0010_0000	0x0013_FFFF	256 KB
HCU_NVR	0x1000_0000	0x1000_03FF	1 KB
OTP_NVR	0x1001_0000	0x1001_03FF	1 KB
CUS_NVR	0x1003_0000	0x1003_03FF	1 KB

HCU 密钥存储单元，可以存储32组128Bits 或 256Bits密钥。安全机制是可写不可读，读操作是将密钥加载到HCU模块。

3. 云途MCAL CryptoDrivers模块的YCT配置

CryptoDrivers有2大主题：加密原语，密钥。配置也是围绕这2大主题展开的。以下是YCT配置CryptoDrivers的界面，按逻辑可以分为三部分：配置加密原语，配置密钥，配置其他。



3.1 配置加密原语

首先配置CryptoPrimitives,如图黄色圈所示，配置加密原语的算法组，算法模式以及提供的加密服务。同时也可以分组管理这些加密原语。



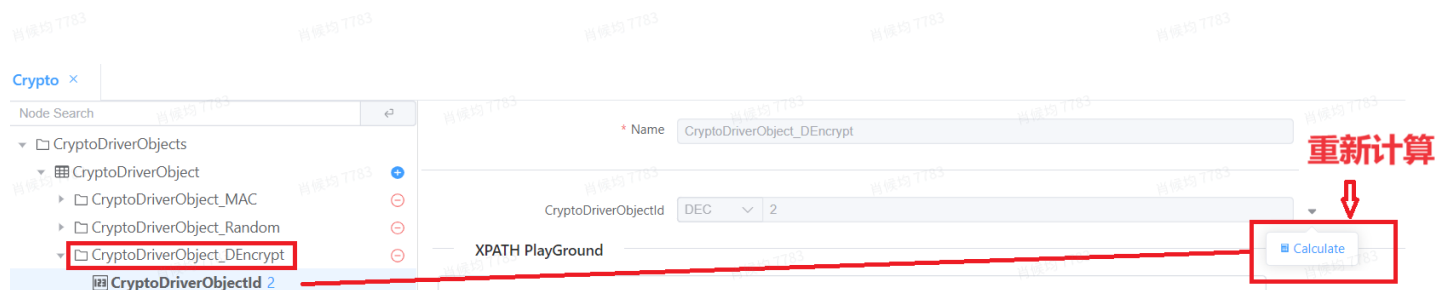
接着配置Objects。Object就是引用不同的加密原语（CryptoPrimitives）



注:

CryptoDriverObjectId区分不同的Object(加密原语), 调用加密服务时需要用到。YCT工具按序号自动管理此参数并生成宏, 方便使用。

若编辑过程中造成序号不连续, 例如删除中间一组配置。可以让**CryptoDriverObjectId**重新计算一下, 方便查看。即便不重新计算也不妨碍代码生成的正确性和使用。

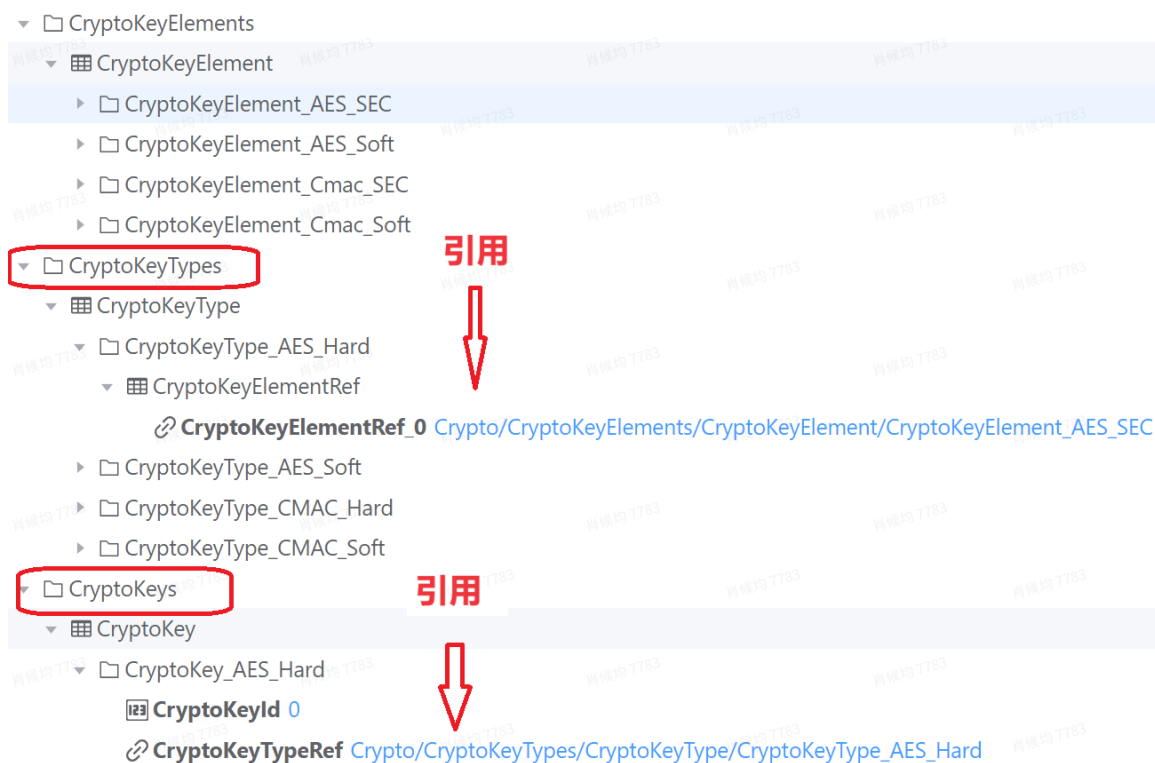


3.2 配置密钥

首先配置**KeyElements**, 主要配置项: Key的长度, 是否使用安全Key, 安全Key或软件Key地址等。Key长度选项**KeyLength_Inbits** 与 **CryptoKeyElementSize**要匹配, 一个是Inbits另一个是Inbytes。若使用安全Key就要选择HCU-NVR槽, 需要先配置Fls模块; 若使用软件Key, 就是配置**KeyRAM_Adress**, 将存放Key的数组名称输入在配置项。其他选项暂时保持默认, 因安全Key对软件不可见, 所有云途的CryptoDrivers暂时不支持Key其他操作。



接着配置CryptoKeyTypes和CryptoKeys。



3.3 配置其他

其他配置项目都在CryptoGeneral里面，包括ErrorDetect,UserMode等。

有3三配置项目一定要注意：

Crypto_KeySlot_BaseRef，引用自Fls，若使用硬件Key，必须配置此选项。

Crypto_DMAInputData_ChannelSelect和**Crypto_DMAOutputData_ChannelSelect**，引用自CddDma，若使用DMA方式，必须配置此选项。



4. 云途MCAL CryptoDrivers模块的主要接口(API)

CryptoDrivers有2大主题：加密原语，密钥。API也是围绕这2大主题实现功能的。因安全密钥（Key）对软件不可见，所有云途的CryptoDrivers暂时不支持对密钥（Key）的操作，与Key操作相关的API均没有实现。

以下是支持的API：

```
/**  
• @brief This interface removes the provided job from the queue and cancels the processing  
of the job if possible.  
• @details  
• @param[in] objectId Holds the identifier of the Crypto Driver Object.  
• @param[in] job Pointer to the configuration of the job. Contains structures with job and  
primitive relevant information.  
• @return Std_ReturnType  
*/  
Std_ReturnType Crypto_CancelJob (uint32 objectId, Crypto_JobType * job);  
  
/**  
• @brief Initializes the Crypto Driver.
```

- @details
- @param[in] configPtr Pointer to a selected configuration structure
- @return void

*/

void **Crypto_Init** (const Crypto_ConfigType * configPtr);

/**

- @brief If asynchronous job processing is configured and there are job queues, the function is called cyclically to process queued jobs.

- @details
- @return void

*/

void **Crypto_MainFunction** (void);

/**

- @brief Performs the crypto primitive, that is configured in the job parameter.
- @details
- @param[in] objectId Holds the identifier of the Crypto Driver Object.
- @param[inout] job Pointer to the configuration of the job. Contains structures with job and primitive relevant information but also pointer to result buffers.
- @return Std_ReturnType

*/

Std_ReturnType **Crypto_ProcessJob** (uint32 objectId, Crypto_JobType * job);

/**

- @brief Returns the version information of this module.
- @details*
- @param[out] versioninfo Pointer to where to store version information of this module

*

- @return void

*/

```
void Crypto_GetVersionInfo(Std_VersionInfoType * versioninfo);
```

5. 云途MCAL CryptoDrivers模块的使用Tips和FAQ

5.1 数据对齐

输入输出数据的长度限制，密钥长度的限制跟不同MCU的HCU模块相关，使用前应查看云途提供的数据手册。现以云途车规MCU YTM32B1ME0的HCU模块为例，谈一下数据对齐的问题。

1、密钥长度

支持密钥长度：128-bits,192-bits,256-bits。

2、输入输出数据

2.1 输入输出数据必须是16 字节对齐。

2.2 HASH SHA256 模式，输入数据最少64字节，输出最多32字节。

2.3 SHA384模式，输入数据至少128字节，输出最多48字节。

2.4 随机数产生，输出数据最多32字节。

2.5 加密/解密模式，输入数据与输出数据必须一致。

5.2 截短输出

HASH（SHA256,SHA384）,CMAC,随机数产生等服务支持截短输出功能。

这几种服务HCU产生的输出数据长度是固定的，实际应用中可能不需要这么多输出结果，可以使用截短输出功能。使用方法很简单，就是将对应Job中jobPrimitiveInputOutput的输出长度改为你需要的长度即可。随机数模块不能通过该方式实现截短功能，随机数必须按照MCU RM中要求的位数设置长度，生成完成后通过软件截取需要的数据。

5.3 数据填充

在使用加密原语服务的时候，有数据对齐的要求。经常会碰到数据长度不符合要求需要填充的情况。有人认为某种算法填充某个值不影响计算结果，这是不正确的认知。数据填充时，可以填充0x00,也可以填充0xFF,可以填充任何值，填充后运算结果是不一样的，也不可能一样。

总结

参考资料

1. [YTM32B1ME0x Reference Manual, REV. 1.3, March 2023](#)
2. [AUTOSAR_SWS_MCUDriver.pdf: Specification of MCU Driver AUTOSAR CP Release 4.4.0](#)
3. [AUTOSAR_SRS_MCUDriver.pdf: Requirements of MCU Driver AUTOSAR CP Release 4.4.0](#)