

MICROSAR CRY DRIVER

Technical Reference

DrvCry_Rh850lcum

Version 1.0

Authors	Tobias Finke
Status	Released

Document Information

History

Author	Date	Version	Remarks
Tobias Finke	2016-11-17	1.00.00	Initial Version of MICROSAR CRY DRIVER

Reference Documents

No.	Source	Title	Version
[1]	AUTOSAR	AUTOSAR_SWS_CryptoServiceManager.pdf	1.2.0
[2]	AUTOSAR	AUTOSAR_TR_BSWModuleList.pdf	1.6.0
[3]	HIS	SHE - Functional Specification	1.1
[4]	Renesas	ICU-M Firmware User's Manual Security Services	Draft 6
[5]	Renesas	ICU-M Firmware User's Manual Installation Guide	Draft 3

**Caution**

This symbol calls your attention to warnings.

**Caution**

We have configured the programs in accordance with your specifications in the questionnaire. Whereas the programs do support other configurations than the one specified in your questionnaire, Vector's release of the programs delivered to your company is expressly restricted to the configuration you have specified in the questionnaire.

Contents

1	Component History	8
2	Introduction.....	9
2.1	Architecture Overview	9
3	Functional Description	11
3.1	Features	11
3.2	Initialization	11
3.3	Main Functions	12
3.4	Key Handling	12
3.5	Key Mapping	12
3.6	Key Update	13
3.7	Abortion of Services	13
3.8	Error Handling	14
3.8.1	Development Error Reporting	14
3.8.2	Production Code Error Reporting	14
4	Integration.....	15
4.1	Scope of Delivery	15
4.1.1	Static Files	15
4.1.2	Dynamic Files	16
4.2	Include Structure	17
4.3	Compiler Abstraction and Memory Mapping	17
4.4	Critical Sections	18
5	API Description.....	19
5.1	Interfaces Overview	19
5.2	Type Definitions	19
5.3	Structures	19
5.3.1	Cry_30_Rh850Icum_Aes128ConfigType	19
5.3.2	Cry_30_Rh850Icum_RngConfigType	19
5.3.3	Cry_30_Rh850Icum_CmacAes128GenConfigType	20
5.3.4	Cry_30_Rh850Icum_CmacAes128VerConfigType	20
5.3.5	Cry_30_Rh850Icum_KeyExtractConfigType	20
5.3.6	Cry_30_Rh850Icum_KeyWrapSymConfigType	21
5.4	Services provided by CRY_30_RH850ICUM	21
5.4.1	Cry_30_Rh850Icum_Init	21
5.4.2	Cry_30_Rh850Icum_InitMemory	22
5.4.3	Cry_30_Rh850Icum_GetVersionInfo	22

5.4.4	Cry_30_Rh850Icum_AesEncrypt128Start.....	23
5.4.5	Cry_30_Rh850Icum_AesEncrypt128Update.....	24
5.4.6	Cry_30_Rh850Icum_AesEncrypt128Finish.....	25
5.4.7	Cry_30_Rh850Icum_AesEncrypt128MainFunction.....	26
5.4.8	Cry_30_Rh850Icum_AesDecrypt128Start.....	27
5.4.9	Cry_30_Rh850Icum_AesDecrypt128Update.....	28
5.4.10	Cry_30_Rh850Icum_AesDecrypt128Finish.....	29
5.4.11	Cry_30_Rh850Icum_AesDecrypt128MainFunction.....	30
5.4.12	Cry_30_Rh850Icum_RngSeedStart.....	31
5.4.13	Cry_30_Rh850Icum_RngSeedUpdate.....	32
5.4.14	Cry_30_Rh850Icum_RngSeedFinish.....	32
5.4.15	Cry_30_Rh850Icum_RngSeedMainFunction.....	33
5.4.16	Cry_30_Rh850Icum_RngGenerate.....	34
5.4.17	Cry_30_Rh850Icum_RngGenerateMainFunction.....	35
5.4.18	Cry_30_Rh850Icum_CmacAes128GenStart.....	36
5.4.19	Cry_30_Rh850Icum_CmacAes128GenUpdate.....	37
5.4.20	Cry_30_Rh850Icum_CmacAes128GenFinish.....	38
5.4.21	Cry_30_Rh850Icum_CmacAes128GenMainFunction.....	39
5.4.22	Cry_30_Rh850Icum_CmacAes128VerStart.....	40
5.4.23	Cry_30_Rh850Icum_CmacAes128VerUpdate.....	41
5.4.24	Cry_30_Rh850Icum_CmacAes128VerFinish.....	42
5.4.25	Cry_30_Rh850Icum_CmacAes128VerMainFunction.....	43
5.4.26	Cry_30_Rh850Icum_KeyExtractStart.....	44
5.4.27	Cry_30_Rh850Icum_KeyExtractUpdate.....	45
5.4.28	Cry_30_Rh850Icum_KeyExtractFinish.....	46
5.4.29	Cry_30_Rh850Icum_KeyExtractMainFunction.....	47
5.4.30	Cry_30_Rh850Icum_KeyWrapSymStart.....	48
5.4.31	Cry_30_Rh850Icum_KeyWrapSymUpdate.....	49
5.4.32	Cry_30_Rh850Icum_KeyWrapSymFinish.....	50
5.4.33	Cry_30_Rh850Icum_KeyWrapSymMainFunction.....	50
5.5	Services used by CRY_30_RH850ICUM.....	51
5.6	Service Ports.....	51

6	Configuration.....	52
6.1	Configuration Variants.....	52
6.2	Configuration with DaVinci Configurator 5.....	52
6.2.1	Common Properties.....	52
6.2.2	AES Encrypt Properties.....	52
6.2.3	AES Decrypt Properties.....	52
6.2.4	CMAC AES-128 Verification Properties.....	53
6.2.5	CMAC AES-128 Generation Properties.....	53

6.2.6	Key Extract Properties	53
6.2.7	Key Wrap Sym Properties	53
6.3	Deviations	53
6.4	Additions/ Extensions.....	54
6.5	Limitations.....	54
6.5.1	Support of Cryptographic Services	54
6.5.2	Tool supported configuration	54
6.5.3	Parallel Access to Services	54
7	Glossary and Abbreviations	55
7.1	Glossary	55
7.2	Abbreviations	55
8	Contact.....	56

Illustrations

Figure 2-1	AUTOSAR 4.x Architecture Overview	9
Figure 2-2	Interfaces to adjacent modules of the CRY_30_RH850ICUM	10
Figure 4-1	Include structure	17

Tables

Table 1-1	Component history.....	8
Table 3-1	Supported AUTOSAR standard conform features	11
Table 3-2	Mapping of KeyId to SHE Keyslots	13
Table 4-1	Static files	16
Table 4-2	Generated files	16
Table 4-3	Compiler abstraction and memory mapping.....	18
Table 5-1	Cry_30_Rh850Icum_Aes128ConfigType	19
Table 5-2	Cry_30_Rh850Icum_RngConfigType	19
Table 5-3	Cry_30_Rh850Icum_CmacAes128GenConfigType	20
Table 5-4	Cry_30_Rh850Icum_CmacAes128VerConfigType	20
Table 5-5	Cry_30_Rh850Icum_KeyExtractConfigType	20
Table 5-6	Cry_30_Rh850Icum_KeyWrapSymConfigType	21
Table 5-7	Cry_30_Rh850Icum_Init	21
Table 5-8	Cry_30_Rh850Icum_InitMemory	22
Table 5-9	Cry_30_Rh850Icum_GetVersionInfo	22
Table 5-10	Cry_30_Rh850Icum_AesEncrypt128Start	23
Table 5-11	Cry_30_Rh850Icum_AesEncrypt128Update.....	24
Table 5-12	Cry_30_Rh850Icum_AesEncrypt128Finish	25
Table 5-13	Cry_30_Rh850Icum_AesEncrypt128MainFunction.....	26
Table 5-14	Cry_30_Rh850Icum_AesDecrypt128Start	27
Table 5-15	Cry_30_Rh850Icum_AesDecrypt128Update	28
Table 5-16	Cry_30_Rh850Icum_AesDecrypt128Finish	29
Table 5-17	Cry_30_Rh850Icum_AesDecrypt128MainFunction.....	30
Table 5-18	Cry_30_Rh850Icum_RngSeedStart.....	31
Table 5-19	Cry_30_Rh850Icum_RngSeedUpdate.....	32
Table 5-20	Cry_30_Rh850Icum_RngSeedFinish.....	32
Table 5-21	Cry_30_Rh850Icum_RngSeedMainFunction	33
Table 5-22	Cry_30_Rh850Icum_RngGenerate.....	34
Table 5-23	Cry_30_Rh850Icum_RngGenerateMainFunction	35
Table 5-24	Cry_30_Rh850Icum_CmacAes128GenStart	36
Table 5-25	Cry_30_Rh850Icum_CmacAes128GenUpdate.....	37
Table 5-26	Cry_30_Rh850Icum_CmacAes128GenFinish.....	38
Table 5-27	Cry_30_Rh850Icum_CmacAes128GenMainFunction.....	39
Table 5-28	Cry_30_Rh850Icum_CmacAes128VerStart.....	40
Table 5-29	Cry_30_Rh850Icum_CmacAes128VerUpdate	41
Table 5-30	Cry_30_Rh850Icum_CmacAes128VerFinish	42
Table 5-31	Cry_30_Rh850Icum_CmacAes128VerMainFunction	43
Table 5-32	Cry_30_Rh850Icum_KeyExtractStart	44
Table 5-33	Cry_30_Rh850Icum_KeyExtractUpdate	45
Table 5-34	Cry_30_Rh850Icum_KeyExtractFinish	46
Table 5-35	Cry_30_Rh850Icum_KeyExtractMainFunction.....	47
Table 5-36	Cry_30_Rh850Icum_KeyWrapSymStart.....	48
Table 5-37	Cry_30_Rh850Icum_KeyWrapSymUpdate	49
Table 5-38	Cry_30_Rh850Icum_KeyWrapSymFinish.....	50
Table 5-39	Cry_30_Rh850Icum_KeyWrapSymMainFunction	51

Table 5-40	Services used by the CRY_30_RH850ICUM	51
Table 6-1	Common configuration properties	52
Table 6-2	Configuration properties of AES-128 Encrypt.....	52
Table 6-3	Configuration properties of AES-128 Decrypt.....	52
Table 6-4	Configuration properties of CMAC AES-128 Verification	53
Table 6-5	Configuration properties of CMAC AES-128 Generation.....	53
Table 6-6	Configuration properties of Key Extract.....	53
Table 6-7	Configuration properties of Key Wrap Sym	53
Table 6-8	Supported AUTOSAR standard conform features	54
Table 7-1	Glossary	55
Table 7-2	Abbreviations.....	55

1 Component History

The component history gives an overview over the important milestones that are supported in the different versions of the component.

Component Version	New Features
1.0	Initial version

Table 1-1 Component history

2 Introduction

This document describes the functionality, API and configuration of the MICROSAR module CRY_30_RH850ICUM as specified in [1].

Supported AUTOSAR Release*:	4	
Supported Configuration Variants:	pre-compile	
Vendor ID:	CRY_VENDOR_ID	30 decimal (= Vector-Informatik, according to HIS)
Module ID:	CRY_MODULE_ID	255 decimal (according to ref. [2])

* For the precise AUTOSAR Release 4.x please see the release specific documentation.

The Cryptographic library module (CRY) offers cryptographic primitives. The CRY module is used by the Crypto Service Manager (CSM).

2.1 Architecture Overview

The figure shows the interfaces to adjacent modules of the CRY_30_RH850ICUM. These interfaces are described in chapter 5.

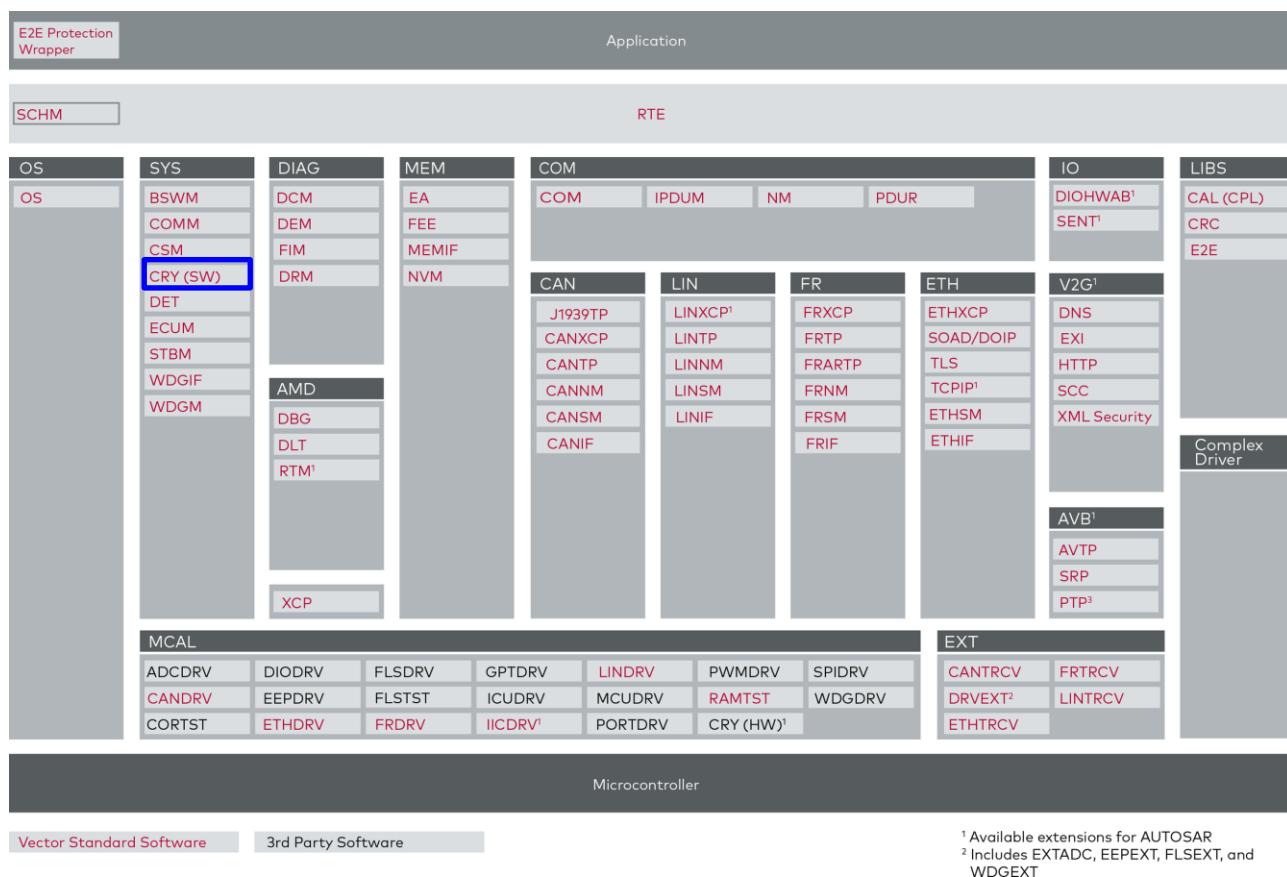


Figure 2-1 AUTOSAR 4.x Architecture Overview

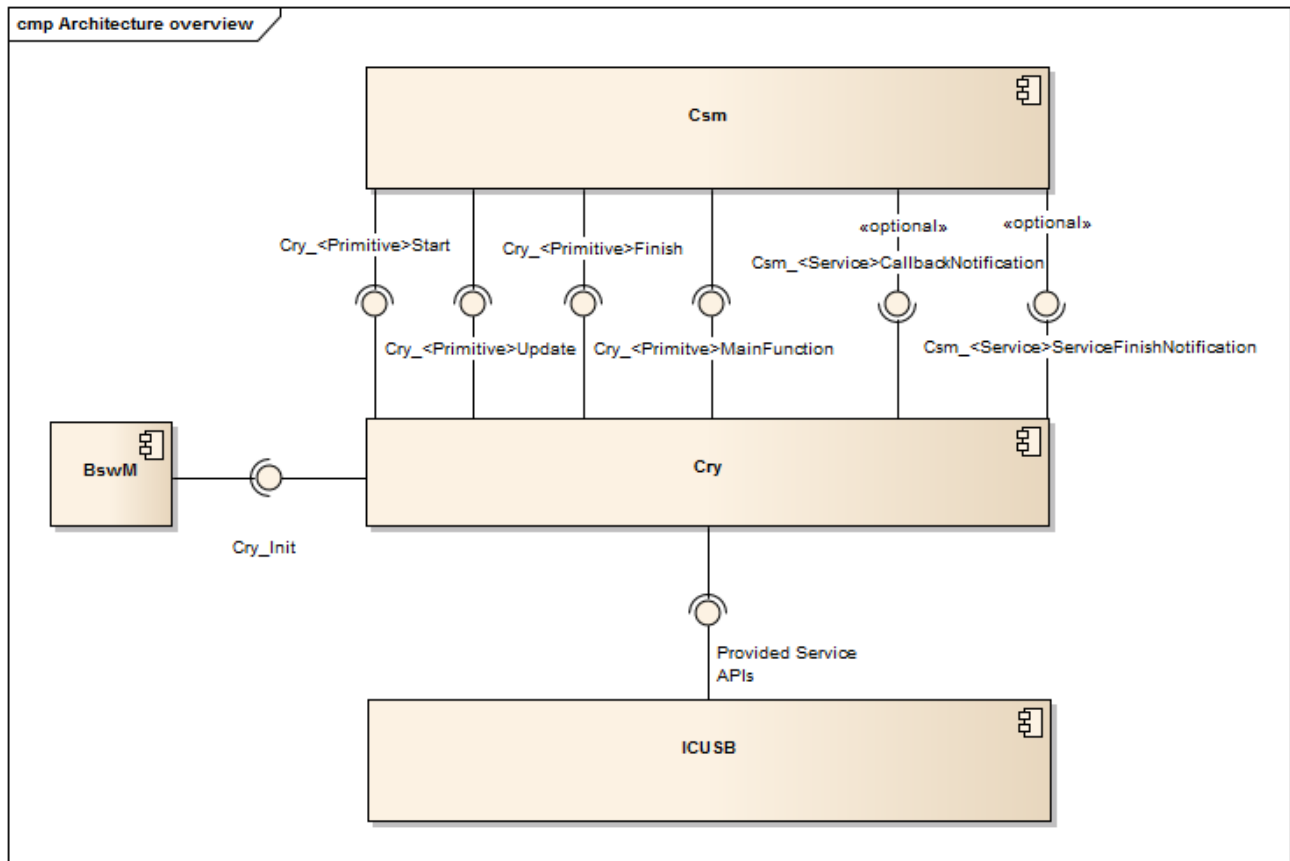


Figure 2-2 Interfaces to adjacent modules of the CRY_30_RH850ICUM

3 Functional Description

3.1 Features

The features listed in the following tables cover the complete functionality specified for the CRY_30_RH850ICUM.

The AUTOSAR standard functionality is specified in [1], the corresponding features are listed in the tables

> Table 3-1 Supported AUTOSAR standard conform features

The following features specified in [1] are supported:

Supported AUTOSAR Standard Conform Features
Synchronous job processing
Asynchronous job processing
Service for Symmetrical Interface (AES128)
Service for MAC Interface (CMAC)
Service for Random Interface (PRNG)
Service for Symmetrical Key Extract Interface
Service for Symmetrical Key Wrapping Interface

Table 3-1 Supported AUTOSAR standard conform features

The CRY_30_RH850ICUM is a wrapper for the SHE services available in the R_ICUMIF provided by Renesas. For details of the R_ICUMIF refer to [4].

3.2 Initialization

Before calling any other functionality of the Cry module the initialization function `Cry_30_Rh850Icum_Init()` has to be called by the BswM at startup.

The Kohaku firmware has to be written to the flash of the ICUM and the ICUM has to be started prior to initializing the module CRY_30_RH850ICUM. For details on how to enable the ICUM refer to the information provided by Renesas [5].

For API details refer to chapter 5.4.1 'Cry_30_Rh850Icum_Init'.

**Note**

If the Kohaku firmware is delivered in debug mode, we have to wait for the register ICUM_ICU2PES to be initialized by the ICUM before issuing any commands to the firmware.

Therefore execute this loop before calling the function `Cry_30_Rh850Icum_Init()`:

```
do
{
    /* wait until ICUM_ICU2PES is either 0xFFFFFFFF or
       contains an address to the ISD struct */
    status = *(volatile uint32 *) (0xFF1F0010u); /* PRQA S 0303 */ /* MD_MSR_11.3 */
} while (status == 0x00000000u);
```

3.3 Main Functions

The CRY module implementation provides a main function for each service. When the usage of sync job processing is disabled, this main function has to be called from the CSM mainfunction context when the service is active.

For API details refer e.g. to chapter 5.4.7 'Cry_30_Rh850Icum_AesEncrypt128MainFunction'.

3.4 Key Handling

The symmetrical keys used by the Cry module are in the format of `Csm_SymKeyType`. This struct consists of a data pointer and a length. If the length equals 1, the first align type of data represents a keyId which is used to select the key slot of the SHE depending on the configuration parameter `keyIdType`. Otherwise, if the length is 16, the data located at the pointer is loaded as a 128 bit key into the RAM key slot of the SHE. This is handled in the specific start function.

3.5 Key Mapping

Depending on the configuration option `CryKeyIdType`, the keyId is interpreted in two different ways.

	CRY_KEYIDTYPE_RAW	CRY_KEYIDTYPE_MAPPED
KeyId	SHE-Keyslot	
0x00	SECRET_KEY	KEY_RAM
0x01	MASTER_ECU_KEY	KEY_1
0x02	BOOT_MAC_KEY	KEY_2
0x03	BOOT_MAC	KEY_3
0x04	KEY_1	KEY_4
0x05	KEY_2	KEY_5

0x06	KEY_3	KEY_6
0x07	KEY_4	KEY_7
0x08	KEY_5	KEY_8
0x09	KEY_6	KEY_9
0x0A	KEY_7	KEY_10
0x0B	KEY_8	KEY_11
0x0C	KEY_9	KEY_12
0x0D	KEY_10	KEY_13
0x0E	KEY_RAM	KEY_14
0x0F	KEY_11	KEY_15
0x10	KEY_12	KEY_16
0x11	KEY_13	KEY_17
0x12	KEY_14	KEY_18
0x13	KEY_15	KEY_19
0x14	KEY_16	KEY_20
0x15	KEY_17	MASTER_ECU_KEY
0x16	KEY_18	-
0x17	KEY_19	-
0x18	KEY_20	-

Table 3-2 Mapping of KeyId to SHE Keyslots

3.6 Key Update

For updating a key slot of the SHE as specified in [3] and [4], the `Cry_30_Rh850Icum_KeyExtract` service is used.

The `dataPtr` in `Cry_30_Rh850Icum_KeyExtractUpdate()` points to an array which consists of the concatenation of a keyID (1 byte) and the three messages M1 (16 byte), M2 (32 byte) and M3 (16 byte). If `dataLength` is 65, these three messages are written to the SHE. The SHE updates the key slot with the key data. Information like Slot ID and the plaintext key data are encoded in the messages M1 to M3.

After writing M1, M2 and M3 to the SHE, the SHE generates M4 and M5 which can be used to verify the key update procedure. In order to retrieve M4 and M5, the `keyPtr` of `Cry_30_Rh850Icum_KeyExtractFinish()` is used to store the 48 bytes of data. Ensure that `CsmSymKeyExtractMaxKeySize` in the CSM configuration is set to at least 48 Bytes.

3.7 Abortion of Services

After every call of `Cry_30_Rh850Icum_<Primitive>Finish()` the SHE is put into an idle-state and any command is aborted if it's still executing. A call of `Cry_30_Rh850Icum_CmacAes128GenFinish()` would e.g. abort a currently running AES decryption. For details refer to 6.5.3 'Parallel Access to Services'.

3.8 Error Handling

3.8.1 Development Error Reporting

The current implementation of the CRY_30_RH850ICUM module does not report any development errors.

3.8.2 Production Code Error Reporting

The current implementation of the CRY_30_RH850ICUM module does not report any production errors.

4 Integration

This chapter gives necessary information for the integration of the MICROSAR CRY_30_RH850ICUM into an application environment of an ECU.

4.1 Scope of Delivery

The delivery of the CRY_30_RH850ICUM contains the files which are described in the chapters 4.1.1 and 4.1.2.

4.1.1 Static Files

File Name	Source Code Delivery	Library Delivery	Description
Cry_30_Rh850Icum.c	■		Source file of the Cry_30_Rh850Icum.
Cry_30_Rh850Icum.h	■		Header file of the Cry_30_Rh850Icum.
Cry_30_Rh850Icum_AesDecrypt128.c	■		Source file of the service Cry_30_Rh850Icum_AesDecrypt128.
Cry_30_Rh850Icum_AesDecrypt128.h	■		Header file of the service Cry_30_Rh850Icum_AesDecrypt128.
Cry_30_Rh850Icum_AesEncrypt128.c	■		Source file of the service Cry_30_Rh850Icum_AesEncrypt128.
Cry_30_Rh850Icum_AesEncrypt128.h	■		Header file of the service Cry_30_Rh850Icum_AesEncrypt128.
Cry_30_Rh850Icum_Rng.c	■		Source file of the service Cry_30_Rh850Icum_Rng.
Cry_30_Rh850Icum_Rng.h	■		Header file of the service Cry_30_Rh850Icum_Rng.
Cry_30_Rh850Icum_CmacAes128Gen.c	■		Source file of the service Cry_30_Rh850Icum_CmacAes128Gen.
Cry_30_Rh850Icum_CmacAes128Gen.h	■		Header file of the Cry_30_Rh850Icum_CmacAes128Gen.
Cry_30_Rh850Icum_CmacAes128Ver.c	■		Source file of the service Cry_30_Rh850Icum_CmacAes128Ver.
Cry_30_Rh850Icum_CmacAes128Ver.h	■		Header file of the service Cry_30_Rh850Icum_CmacAes128Ver.

File Name	Source Code Delivery	Library Delivery	Description
Cry_30_Rh850lcum_KeyExtract.c	■		Source file of the service Cry_30_Rh850lcum_KeyExtract.
Cry_30_Rh850lcum_KeyExtract.h	■		Header file of the service Cry_30_Rh850lcum_KeyExtract.
Cry_30_Rh850lcum_KeyWrapSym.c	■		Source file of the service Cry_30_Rh850lcum_KeyWrapSym.
Cry_30_Rh850lcum_KeyWrapSym.h	■		Header file of the service Cry_30_Rh850lcum_KeyWrapSym.

Table 4-1 Static files

4.1.2 Dynamic Files

The dynamic files are generated with the help of Cfg5.

File Name	Description
Cry_30_Rh850lcum_Cfg.c	This is the configuration source file.
Cry_30_Rh850lcum_Cfg.h	This is the configuration header file.

Table 4-2 Generated files

4.2 Include Structure

Figure 4-1 shows the include structure of the Cry. Some includes are optional and depend on the configuration. `Cry_30_Rh850Icum_<Primitive>.h` stands for every used cryptographic primitive.

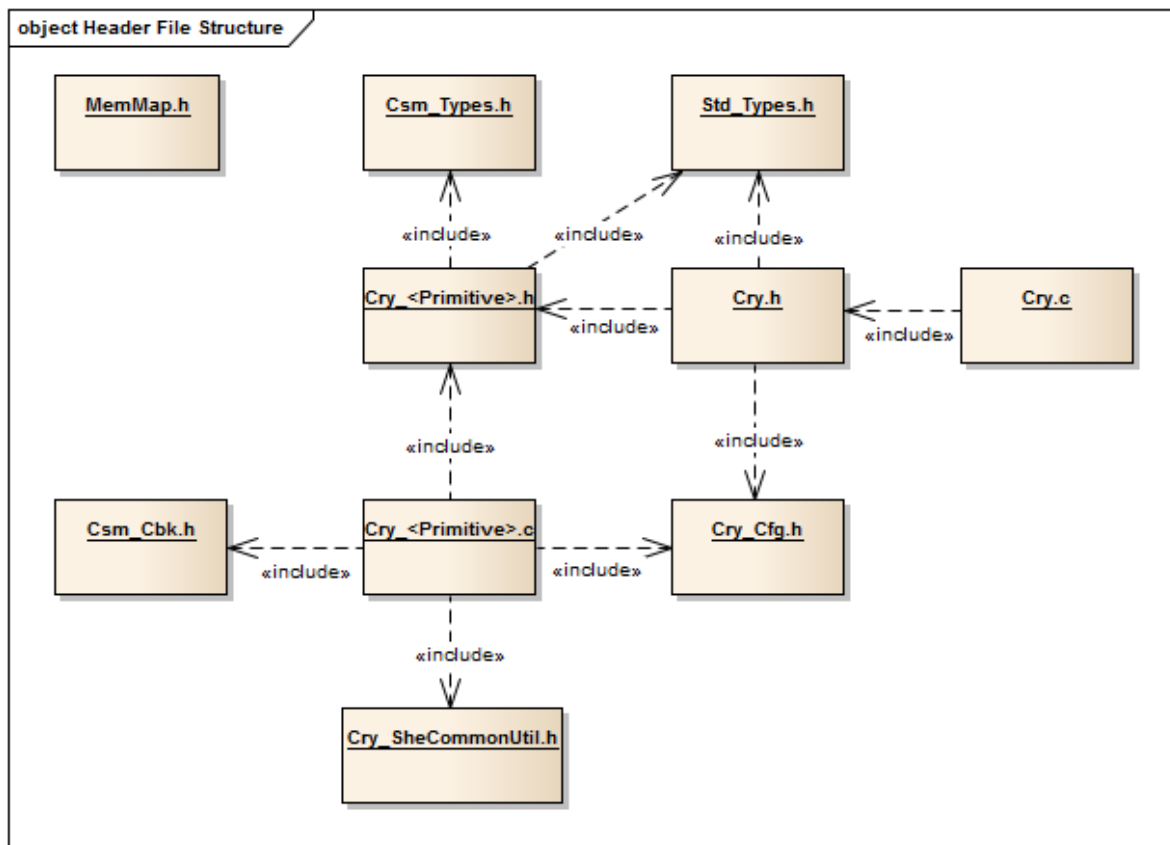


Figure 4-1 Include structure

4.3 Compiler Abstraction and Memory Mapping

The objects (e.g. variables, functions, constants) are declared by compiler independent definitions – the compiler abstraction definitions. Each compiler abstraction definition is assigned to a memory section.

The following table (Table 4-3) contains the memory section names and the compiler abstraction definitions of the CRY_30_RH850ICUM and illustrates their assignment among each other.

Memory Mapping Sections	Compiler Abstraction Definitions		
	CRY_30_RH850ICUM_CODE	CRY_30_RH850ICUM_VAR_NOINIT	CRY_30_RH850ICUM_APPL_VAR
	CRY_30_RH850ICUM_START_SEC_CODE CRY_30_RH850ICUM_STOP_SEC_CODE	■	■
	CRY_30_RH850ICUM_START_SEC_VAR_NOINIT_8BIT CRY_30_RH850ICUM_STOP_SEC_VAR_NOINIT_8BIT	■	
CRY_30_RH850ICUM_START_SEC_VAR_NOINIT_UN NSPECIFIED CRY_30_RH850ICUM_STOP_SEC_VAR_NOINIT_UN SPECIFIED		■	

Table 4-3 Compiler abstraction and memory mapping

4.4 Critical Sections

The current implementation of the CRY module does not have any critical section.

5 API Description

5.1 Interfaces Overview

For an interfaces overview please see Figure 2-2.

5.2 Type Definitions

The types defined by the CRY_30_RH850ICUM are described in this chapter.

5.3 Structures

5.3.1 Cry_30_Rh850Icum_Aes128ConfigType

This structure represents the configuration for the Cry_30_Rh850Icum_AesDecrypt128 and Cry_30_Rh850Icum_AesEncrypt128 service

Struct Element Name	C-Type	Description	Value Range
buffer	Cry_30_Rh850Icum_Aes128WorkSpaceType*	Pointer to a provided buffer which will be used as workspace for the primitives	
blockMode	uint8	Block mode	CRY_BLOCKMODE_ECB, CRY_BLOCKMODE_CBC
keyIdType	uint8	Defines the interpretation of the keyid	CRY_KEYIDTYPE_MAPPED, CRY_KEYIDTYPE_RAW

Table 5-1 Cry_30_Rh850Icum_Aes128ConfigType

5.3.2 Cry_30_Rh850Icum_RngConfigType

This structure represents the configuration for the Cry_30_Rh850Icum_Rng service

Struct Element Name	C-Type	Description	Value Range
buffer	Cry_30_Rh850Icum_RngWorkSpaceType*	Pointer to a provided buffer which will be used as workspace for the primitives	

Table 5-2 Cry_30_Rh850Icum_RngConfigType

5.3.3 Cry_30_Rh850Icum_CmacAes128GenConfigType

This structure represents the configuration for the Cry_30_Rh850Icum_CmacAes128Gen service

Struct Element Name	C-Type	Description	Value Range
buffer	Cry_30_Rh850Icum_CmacAes128GenWorkSpaceType *	Pointer to a provided buffer which will be used as workspace for the primitives	
keyIdType	uint8	Defines the interpretation of the keyid	CRY_KEYIDTYPE_MAPPED, CRY_KEYIDTYPE_RAW

Table 5-3 Cry_30_Rh850Icum_CmacAes128GenConfigType

5.3.4 Cry_30_Rh850Icum_CmacAes128VerConfigType

This structure represents the configuration for the Cry_30_Rh850Icum_CmacAes128Ver service

Struct Element Name	C-Type	Description	Value Range
buffer	Cry_30_Rh850Icum_CmacAes128VerWorkSpaceType *	Pointer to a provided buffer which will be used as workspace for the primitives	
keyIdType	uint8	Defines the interpretation of the keyid	CRY_KEYIDTYPE_MAPPED, CRY_KEYIDTYPE_RAW
lengthInBytes	uint8	Defines if Mac Length is interpreted in bytes or bits	CRY_MAC_LENGTH_IN_BYTES, CRY_MAC_LENGTH_IN_BITS

Table 5-4 Cry_30_Rh850Icum_CmacAes128VerConfigType

5.3.5 Cry_30_Rh850Icum_KeyExtractConfigType

This structure represents the configuration for the Cry_30_Rh850Icum_KeyExtract service

Struct Element Name	C-Type	Description	Value Range
buffer	Cry_30_Rh850Icum_KeyExtractWorkSpaceType *	Pointer to a provided buffer which will be used as workspace for the primitives	
keyIdType	uint8	Defines the interpretation of the keyid	CRY_KEYIDTYPE_MAPPED, CRY_KEYIDTYPE_RAW

Table 5-5 Cry_30_Rh850Icum_KeyExtractConfigType

5.3.6 Cry_30_Rh850Icum_KeyWrapSymConfigType

This structure represents the configuration for the Cry_30_Rh850Icum_KeyWrapSym service

Struct Element Name	C-Type	Description	Value Range
buffer	Cry_30_Rh850Icum_KeyWrapSymWorkspaceType *	Pointer to a provided buffer which will be used as workspace for the primitives	
keyIdType	uint8	Defines the interpretation of the keyid	CRY_KEYIDTYPE_MAPPED, CRY_KEYIDTYPE_RAW

Table 5-6 Cry_30_Rh850Icum_KeyWrapSymConfigType

5.4 Services provided by CRY_30_RH850ICUM

5.4.1 Cry_30_Rh850Icum_Init

Prototype	
void Cry_30_Rh850Icum_Init (void)	
Parameter	
-	
Return code	
-	
Functional Description	
This function initializes the Cry.	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function is synchronous. > This function is non-reentrant. > This function has to be called during start-up. 	
Call Context	
<ul style="list-style-type: none"> > This function can be called from task level only. 	

Table 5-7 Cry_30_Rh850Icum_Init

5.4.2 Cry_30_Rh850Icum_InitMemory

Prototype	
void Cry_30_Rh850Icum_InitMemory (void)	
Parameter	
-	
Return code	
-	
Functional Description	
This function is currently empty but required by the MICROSAR stack.	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function is synchronous. > This function is non-reentrant. 	
Call Context	
<ul style="list-style-type: none"> > This function can be called from task level only. 	

Table 5-8 Cry_30_Rh850Icum_InitMemory

5.4.3 Cry_30_Rh850Icum_GetVersionInfo

Prototype	
void Cry_30_Rh850Icum_GetVersionInfo (Std_VersionInfoType *cryVerInfoPtr)	
Parameter	
cryVerInfoPtr	Pointer where the version information shall be copied to.
Return code	
-	
Functional Description	
This function copies the Cry version information to the location provided by the pointer.	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function is synchronous. > This function is non-reentrant. > This function is only available if 'Version Info Api' is enabled. 	
Call Context	
<ul style="list-style-type: none"> > This function can be called from task and interrupt level. 	

Table 5-9 Cry_30_Rh850Icum_GetVersionInfo

5.4.4 Cry_30_Rh850Icum_AesEncrypt128Start

Prototype	
Csm_ReturnType Cry_30_Rh850Icum_AesEncrypt128Start (Const void *cfgPtr, const Csm_SymKeyType *keyPtr, const uint8 *InitVectorPtr, uint32 InitVectorLength)	
Parameter	
cfgPtr	Holds a pointer to the configuration of this service. See Cry_30_Rh850Icum_Aes128ConfigType for more information.
keyPtr	Holds a pointer to the key which has to be used during the symmetrical encryption operation.
InitVectorPtr	Holds a pointer to initialization vector which has to be used during the symmetrical encryption.
InitVectorLength	Holds the length of the initialization vector which has to be used during the symmetrical encryption.
Return code	
CSM_E_OK	Request successful.
CSM_E_NOT_OK	Request failed.
CSM_E_BUSY	Request failed, service is still busy
Functional Description	
This interface shall be used to initialize the symmetrical encryption service.	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function can be synchronous or asynchronous. > This function is non-reentrant. > This function is called by application. 	
Call Context	
<ul style="list-style-type: none"> > This function can be called from task level only. 	

Table 5-10 Cry_30_Rh850Icum_AesEncrypt128Start

5.4.5 Cry_30_Rh850Icum_AesEncrypt128Update


Prototype	
Csm_ReturnType Cry_30_Rh850Icum_AesEncrypt128Update (Const void *cfgPtr, const uint8 *plainTextPtr, uint32 plainTextLength, uint8 *cipherTextPtr, uint32 *cipherTextLengthPtr)	
Parameter	
cfgPtr	Holds a pointer to the configuration of this service. See Cry_30_Rh850Icum_Aes128ConfigType for more information.
plainTextPtr	Holds a pointer to the data for which a encrypted text shall be computed.
plainTextLength	Contains the number of bytes for which the encrypted text shall be computed. Only values which are the same or a multiple of the blocksize (16 bytes) are allowed.
cipherTextPtr	Holds a pointer to the memory location which will hold the encrypted text.
cipherTextLengthPtr	Holds a pointer to the memory location in which the length information is stored. On calling this function this parameter shall contain the size of the provided buffer. When the request has finished, the actual length of the returned encrypted text shall be stored.
Return code	
CSM_E_OK	Request successful.
CSM_E_NOT_OK	Request failed.
CSM_E_SMALL_BUFFER	The provided buffer is too small to store the result and truncation was not allowed.
Functional Description	
This interface shall be used to feed the symmetrical encryption service with the input data.	
Particularities and Limitations	
<div>  <div> <p>Note</p> <p>It's not possible to call Cry_30_Rh850Icum_AesEncryptUpdate multiple times in order to feed the symmetrical encryption service with separate input data chunks. Therefore plainTextLength must be set to the length of the complete input data. The data has to be padded to the length of the blocksize (16 bytes) or a multiple of it before calling this function.</p> </div> </div> <ul style="list-style-type: none"> > This function can be synchronous or asynchronous. > This function is non-reentrant. > This function is called by application. 	
Call Context	
<ul style="list-style-type: none"> > This function can be called from task level only. 	

Table 5-11 Cry_30_Rh850Icum_AesEncrypt128Update

5.4.6 Cry_30_Rh850Icum_AesEncrypt128Finish

Prototype	
Csm_ReturnType Cry_30_Rh850Icum_AesEncrypt128Finish (Const void *cfgPtr, uint8 *cipherTextPtr, uint32 *cipherTextLengthPtr)	
Parameter	
cfgPtr	Holds a pointer to the configuration of this service. See Cry_30_Rh850Icum_Aes128ConfigType for more information.
cipherTextPtr	Holds a pointer to the memory location which will hold the encrypted text.
cipherTextLengthPtr	Holds a pointer to the memory location in which the length information is stored. On calling this function this parameter shall contain the size of the provided buffer. When the request has finished, the actual length of the returned encrypted text shall be stored.
Return code	
CSM_E_OK	Request successful.
CSM_E_NOT_OK	Request failed.
CSM_E_SMALL_BUFFER	The provided buffer is too small to store the result and truncation was not allowed.
Functional Description	
This interface shall be used to finish the symmetrical encryption service.	
Particularities and Limitations	
<ul style="list-style-type: none">> This function can be synchronous or asynchronous.> This function is non-reentrant.> This function is called by application.	
Call Context	
<ul style="list-style-type: none">> This function can be called from task level only.	

Table 5-12 Cry_30_Rh850Icum_AesEncrypt128Finish

5.4.7 Cry_30_Rh850Icum_AesEncrypt128MainFunction


Prototype	
void Cry_30_Rh850Icum_AesEncrypt128MainFunction (void)	
Parameter	
-	
Return code	
-	
Functional Description	
This function implements the asynchronous service handling.	
	Note
	This function is empty if 'Use Sync Job Processing' is enabled.
Particularities and Limitations	
<ul style="list-style-type: none">> This function is synchronous.> This function is not reentrant.> This function has to be called by CSM.> This function must not be called by the application.	
Call Context	
<ul style="list-style-type: none">> This function can be called from task level only.	

Table 5-13 Cry_30_Rh850Icum_AesEncrypt128MainFunction

5.4.8 Cry_30_Rh850Icum_AesDecrypt128Start

Prototype	
Csm_ReturnType Cry_30_Rh850Icum_AesDecrypt128Start (Const void *cfgPtr, const Csm_SymKeyType *keyPtr, const uint8 *InitVectorPtr, uint32 InitVectorLength)	
Parameter	
cfgPtr	Holds a pointer to the configuration of this service. See Cry_30_Rh850Icum_Aes128ConfigType for more information.
keyPtr	Holds a pointer to the key which has to be used during the symmetrical decryption operation.
InitVectorPtr	Holds a pointer to initialization vector which has to be used during the symmetrical decryption.
InitVectorLength	Holds the length of the initialization vector which has to be used during the symmetrical decryption.
Return code	
CSM_E_OK	Request successful.
CSM_E_NOT_OK	Request failed.
CSM_E_BUSY	Request failed, service is still busy.
Functional Description	
This interface shall be used to initialize the symmetrical decryption service of the CSM module.	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function can be synchronous or asynchronous. > This function is non-reentrant. > This function is called by application. 	
Call Context	
<ul style="list-style-type: none"> > This function can be called from task level only. 	

Table 5-14 Cry_30_Rh850Icum_AesDecrypt128Start

5.4.9 Cry_30_Rh850Icum_AesDecrypt128Update


Prototype	
Csm_ReturnType Cry_30_Rh850Icum_AesDecrypt128Update (Const void *cfgPtr, const uint8 *cipherTextPtr, uint32 cipherTextLength, uint8 *plainTextPtr, uint32 *plainTextLengthPtr)	
Parameter	
cfgPtr	Holds a pointer to the configuration of this service. See Cry_30_Rh850Icum_Aes128ConfigType for more information.
cipherTextPtr	Holds a pointer to the data for which a decrypted text shall be computed.
cipherTextLength	Contains the number of bytes for which the decrypted text shall be computed. Only values which are the same or a multiple of the blocksize (16 bytes) are allowed.
plainTextPtr	Holds a pointer to the memory location which will hold the decrypted text.
plainTextLengthPtr	Holds a pointer to the memory location in which the length information is stored. On calling this function this parameter shall contain the size of the provided buffer. When the request has finished, the actual length of the returned decrypted text shall be stored.
Return code	
CSM_E_OK	Request successful.
CSM_E_NOT_OK	Request failed.
CSM_E_SMALL_BUFFER	The provided buffer is too small to store the result and truncation was not allowed.
Functional Description	
This interface shall be used to feed the symmetrical decryption service with the input data.	
Particularities and Limitations	
<div>Note It's not possible to call Cry_30_Rh850Icum_AesDecryptUpdate multiple times in order to feed the symmetrical decryption service with separate input data chunks. Therefore cipherTextLength must be set to the length of the complete input data.</div> <div><ul style="list-style-type: none">> This function can be synchronous or asynchronous.> This function is non-reentrant.> This function is called by application.</div>	
Call Context	
<ul style="list-style-type: none">> This function can be called from task level only.	

Table 5-15 Cry_30_Rh850Icum_AesDecrypt128Update

5.4.10 Cry_30_Rh850Icum_AesDecrypt128Finish

Prototype	
Csm_ReturnType Cry_30_Rh850Icum_AesDecrypt128Finish (Const void *cfgPtr, uint8 *plainTextPtr, uint32 *plainTextLengthPtr)	
Parameter	
cfgPtr	Holds a pointer to the configuration of this service. See Cry_30_Rh850Icum_Aes128ConfigType for more information.
plainTextPtr	Holds a pointer to the memory location which will hold the decrypted text.
plainTextLengthPtr	Holds a pointer to the memory location in which the length information is stored. On calling this function this parameter shall contain the size of the provided buffer. When the request has finished, the actual length of the returned decrypted text shall be stored.
Return code	
CSM_E_OK	Request successful.
CSM_E_NOT_OK	Request failed.
CSM_E_SMALL_BUFFER	The provided buffer is too small to store the result and truncation was not allowed.
Functional Description	
This interface shall be used to finish the symmetrical decryption service.	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function can be synchronous or asynchronous. > This function is non-reentrant. > This function is called by application. 	
Call Context	
<ul style="list-style-type: none"> > This function can be called from task level only. 	

Table 5-16 Cry_30_Rh850Icum_AesDecrypt128Finish

5.4.11 Cry_30_Rh850Icum_AesDecrypt128MainFunction


Prototype	
void Cry_30_Rh850Icum_AesDecrypt128MainFunction (void)	
Parameter	
-	
Return code	
-	
Functional Description	
This function implements the asynchronous service handling.	
	Note
	This function is empty if 'Use Sync Job Processing' is enabled.
Particularities and Limitations	
<ul style="list-style-type: none"> > This function is synchronous. > This function is not reentrant. > This function has to be called by CSM. > This function must not be called by the application. 	
Call Context	
<ul style="list-style-type: none"> > This function can be called from task level only. 	

Table 5-17 Cry_30_Rh850Icum_AesDecrypt128MainFunction

5.4.12 Cry_30_Rh850Icum_RngSeedStart

Prototype	
Csm_ReturnType Cry_30_Rh850Icum_RngSeedStart (Const void *cfgPtr)	
Parameter	
cfgPtr	Holds a pointer to the configuration of this service. See Cry_30_Rh850Icum_RngConfigType for more information.
Return code	
CSM_E_OK	Request successful.
CSM_E_NOT_OK	Request failed.
CSM_E_BUSY	Request failed, service is still busy.
Functional Description	
This function initializes the workspace for the random seed service.	
Particularities and Limitations	
<ul style="list-style-type: none">> This function can be synchronous or asynchronous.> This function is non-reentrant.> This function is called by application.	
Call Context	
<ul style="list-style-type: none">> This function can be called from task level only.	

Table 5-18 Cry_30_Rh850Icum_RngSeedStart

5.4.13 Cry_30_Rh850Icum_RngSeedUpdate

Prototype	
Csm_ReturnType Cry_30_Rh850Icum_RngSeedUpdate (Const void *cfgPtr, const uint8 *seedPtr, uint32 seedLength)	
Parameter	
cfgPtr	Holds a pointer to the configuration of this service. See Cry_30_Rh850Icum_RngConfigType for more information.
seedPtr	Holds a pointer to the seed for the random number generator.
seedLength	Contains the length of the seed in bytes. Only the value 16 is supported.
Return code	
CSM_E_OK	Request successful.
CSM_E_NOT_OK	Request failed.
Functional Description	
This function shall be used to feed a seed to the random number generator.	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function can be synchronous or asynchronous. > This function is non-reentrant. > This function is called by application. 	
Call Context	
<ul style="list-style-type: none"> > This function can be called from task level only. 	

Table 5-19 Cry_30_Rh850Icum_RngSeedUpdate

5.4.14 Cry_30_Rh850Icum_RngSeedFinish

Prototype	
Csm_ReturnType Cry_30_Rh850Icum_RngSeedFinish (Const void *cfgPtr)	
Parameter	
cfgPtr	Holds a pointer to the configuration of this service. See Cry_30_Rh850Icum_RngConfigType for more information.
Return code	
CSM_E_OK	Request successful.
Functional Description	
This function finalizes the random seed service.	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function can be synchronous or asynchronous. > This function is non-reentrant. > This function is called by application. 	
Call Context	
<ul style="list-style-type: none"> > This function can be called from task level only. 	

Table 5-20 Cry_30_Rh850Icum_RngSeedFinish

5.4.15 Cry_30_Rh850lcum_RngSeedMainFunction


Prototype	
void Cry_30_Rh850lcum_RngSeedMainFunction (void)	
Parameter	
-	
Return code	
-	
Functional Description	
This function implements the asynchronous service handling.	
<div>  <div> <p>Note</p> <p>This function is empty if 'Use Sync Job Processing' is enabled.</p> </div> </div>	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function is synchronous. > This function is not reentrant. > This function has to be called by CSM. > This function must not be called by the application. 	
Call Context	
<ul style="list-style-type: none"> > This function can be called from task level only. 	

Table 5-21 Cry_30_Rh850lcum_RngSeedMainFunction

5.4.16 Cry_30_Rh850Icum_RngGenerate

Prototype	
Csm_ReturnType Cry_30_Rh850Icum_RngGenerate (Const void *cfgPtr, uint8 *resultPtr, uint32 resultLength)	
Parameter	
cfgPtr	Holds a pointer to the configuration of this service. See Cry_30_Rh850Icum_RngConfigType for more information.
resultPtr	Holds a pointer to the memory location which will hold the result of the random number generation. The memory location must have at least the size "resultLength".
resultLength	Holds the amount of random bytes which should be generated.
Return code	
CSM_E_OK	Request successful.
CSM_E_NOT_OK	Request failed.
CSM_E_BUSY	Request failed, service is still busy.
Functional Description	
Generates a pseudo random number with the help of the RNG of the SHE.	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function can be synchronous or asynchronous. > This function is non-reentrant. > This function is called by application. 	
Call Context	
<ul style="list-style-type: none"> > This function can be called from task level only. 	

Table 5-22 Cry_30_Rh850Icum_RngGenerate

5.4.17 Cry_30_Rh850Icum_RngGenerateMainFunction


Prototype	
void Cry_30_Rh850Icum_RngGenerateMainFunction (void)	
Parameter	
-	
Return code	
-	
Functional Description	
This function implements the asynchronous service handling.	
	Note
	This function is empty if 'Use Sync Job Processing' is enabled.
Particularities and Limitations	
<ul style="list-style-type: none">> This function is synchronous.> This function is not reentrant.> This function has to be called by CSM.> This function must not be called by the application.	
Call Context	
<ul style="list-style-type: none">> This function can be called from task level only.	

Table 5-23 Cry_30_Rh850Icum_RngGenerateMainFunction

5.4.18 Cry_30_Rh850Icum_CmacAes128GenStart

Prototype	
Csm_ReturnType Cry_30_Rh850Icum_CmacAes128GenStart (Const void *cfgPtr, const Csm_SymKeyType *keyPtr)	
Parameter	
cfgPtr	Holds a pointer to the configuration of this service. See Cry_30_Rh850Icum_CmacAes128GenConfigType for more information.
keyPtr	Holds a pointer to the key necessary for the MAC generation.
Return code	
CSM_E_OK	Request successful.
CSM_E_NOT_OK	Request failed.
CSM_E_BUSY	Request failed, service is still busy.
Functional Description	
This interface shall be used to initialize the SHE-CMAC generation.	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function can be synchronous or asynchronous. > This function is non-reentrant. > This function is called by application. 	
Call Context	
<ul style="list-style-type: none"> > This function can be called from task level only. 	

Table 5-24 Cry_30_Rh850Icum_CmacAes128GenStart

5.4.19 Cry_30_Rh850Icum_CmacAes128GenUpdate


Prototype	
Csm_ReturnType Cry_30_Rh850Icum_CmacAes128GenUpdate (Const void *cfgPtr, const uint8 *dataPtr, uint32 dataLength)	
Parameter	
cfgPtr	Holds a pointer to the configuration of this service. See Cry_30_Rh850Icum_CmacAes128GenConfigType for more information.
dataPtr	Holds a pointer to the data for which a MAC shall be computed.
dataLength	Contains the number of bytes for which the MAC shall be computed.
Return code	
CSM_E_OK	Request successful.
CSM_E_NOT_OK	Request failed.
Functional Description	
This function shall be used to feed the SHE-CMAC generation with input data.	
Particularities and Limitations	
<div>Note Due to limitations in the API of the SHE, it's not possible to call Cry_30_Rh850Icum_CmacAes128GenUpdate multiple times in order to feed the CMAC generation with separate input data chunks. Therefore dataLength must be set to the length of the complete input data.</div> <div><ul style="list-style-type: none">> This function can be synchronous or asynchronous.> This function is non-reentrant.> This function is called by application.</div>	
Call Context	
<ul style="list-style-type: none">> This function can be called from task level only.	

Table 5-25 Cry_30_Rh850Icum_CmacAes128GenUpdate

5.4.20 Cry_30_Rh850Icum_CmacAes128GenFinish

Prototype	
Csm_ReturnType Cry_30_Rh850Icum_CmacAes128GenFinish (Const void *cfgPtr, const uint8 *resultPtr, uint32* resultLengthPtr, boolean TruncationIsAllowed)	
Parameter	
cfgPtr	Holds a pointer to the configuration of this service. See Cry_30_Rh850Icum_CmacAes128GenConfigType for more information.
resultPtr	Holds a pointer to the memory location which will hold the result of the MAC generation. If the result does not fit into the given buffer, and truncation is allowed, the result shall be truncated
resultLengthPtr	Holds a pointer to the memory location in which the length information is stored. On calling this function this parameter shall contain the size of the buffer provided by resultPtr. When the request has finished, the actual length of the returned MAC shall be stored.
TruncationIsAllowed	This parameter states whether a truncation of the result is allowed or not. TRUE: truncation is allowed. FALSE: truncation is not allowed.
Return code	
CSM_E_OK	Request successful.
CSM_E_NOT_OK	Request failed
CSM_E_SMALL_BUFFER	The provided buffer is too small to store the result, and truncation was not allowed.
Functional Description	
This interface shall be used to finish the SHE-CMAC generation.	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function can be synchronous or asynchronous. > This function is non-reentrant. > This function is called by application. 	
Call Context	
<ul style="list-style-type: none"> > This function can be called from task level only. 	

Table 5-26 Cry_30_Rh850Icum_CmacAes128GenFinish

5.4.21 Cry_30_Rh850Icum_CmacAes128GenMainFunction


Prototype	
void Cry_30_Rh850Icum_CmacAes128GenMainFunction (void)	
Parameter	
-	
Return code	
-	
Functional Description	
This function implements the asynchronous service handling.	
	Note
	This function is empty if 'Use Sync Job Processing' is enabled.
Particularities and Limitations	
<ul style="list-style-type: none">> This function is synchronous.> This function is not reentrant.> This function has to be called by CSM.> This function must not be called by the application.	
Call Context	
<ul style="list-style-type: none">> This function can be called from task level only.	

Table 5-27 Cry_30_Rh850Icum_CmacAes128GenMainFunction

5.4.22 Cry_30_Rh850Icum_CmacAes128VerStart

Prototype	
Csm_ReturnType Cry_30_Rh850Icum_CmacAes128VerStart (Const void *cfgPtr, const Csm_SymKeyType *keyPtr)	
Parameter	
cfgPtr	Holds a pointer to the configuration of this service. See Cry_30_Rh850Icum_CmacAes128GenConfigType for more information.
keyPtr	Holds a pointer to the key necessary for the MAC verification.
Return code	
CSM_E_OK	Request successful.
CSM_E_NOT_OK	Request failed.
CSM_E_BUSY	Request failed, service is still busy.
Functional Description	
This interface shall be used to initialize the SHE-CMAC verification.	
Particularities and Limitations	
<ul style="list-style-type: none">> This function can be synchronous or asynchronous.> This function is non-reentrant.> This function is called by application.	
Call Context	
<ul style="list-style-type: none">> This function can be called from task level only.	

Table 5-28 Cry_30_Rh850Icum_CmacAes128VerStart

5.4.23 Cry_30_Rh850Icum_CmacAes128VerUpdate


Prototype	
Csm_ReturnType Cry_30_Rh850Icum_CmacAes128VerUpdate (Const void *cfgPtr, const uint8 *dataPtr, uint32 dataLength)	
Parameter	
cfgPtr	Holds a pointer to the configuration of this service. See Cry_30_Rh850Icum_CmacAes128GenConfigType for more information.
dataPtr	Holds a pointer to the data for which a MAC shall be verified.
dataLength	Contains the number of bytes for which the MAC shall be verified.
Return code	
CSM_E_OK	Request successful.
CSM_E_NOT_OK	Request failed.
Functional Description	
This function shall be used to feed the SHE-CMAC verification with the input data.	
Particularities and Limitations	
<div>Note Due to limitations in the API of the SHE, it's not possible to call Cry_30_Rh850Icum_CmacAes128VerUpdate multiple times in order to feed the CMAC verification with separate input data chunks. Therefore dataLength must be set to the length of the complete input data.</div> <div><ul style="list-style-type: none">> This function can be synchronous or asynchronous.> This function is non-reentrant.> This function is called by application.</div>	
Call Context	
<ul style="list-style-type: none">> This function can be called from task level only.	

Table 5-29 Cry_30_Rh850Icum_CmacAes128VerUpdate

5.4.24 Cry_30_Rh850Icum_CmacAes128VerFinish

Prototype	
Csm_ReturnType Cry_30_Rh850Icum_CmacAes128VerFinish (Const void *cfgPtr, const uint8 *MacPtr, uint32 MacLength, Csm_VerifyResultType *resultPtr)	
Parameter	
cfgPtr	Holds a pointer to the configuration of this service. See Cry_30_Rh850Icum_CmacAes128GenConfigType for more information.
MacPtr	Holds a pointer to the memory location which will hold the MAC to verify.
MacLength	Holds the length of the MAC to be verified. Depending on the configuration, this value is interpreted as number of bits or number of bytes to verify.
resultPtr	Holds a pointer to the memory location which will hold the MAC.
Return code	
CSM_E_OK	Request successful.
CSM_E_NOT_OK	Request failed
Functional Description	
This interface shall be used to finish the SHE-CMAC verification.	
Particularities and Limitations	
<ul style="list-style-type: none">> This function can be synchronous or asynchronous.> This function is non-reentrant.> This function is called by application.	
Call Context	
<ul style="list-style-type: none">> This function can be called from task level only.	

Table 5-30 Cry_30_Rh850Icum_CmacAes128VerFinish

5.4.25 Cry_30_Rh850Icum_CmacAes128VerMainFunction


Prototype	
void Cry_30_Rh850Icum_CmacAes128VerMainFunction (void)	
Parameter	
-	
Return code	
-	
Functional Description	
This function implements the asynchronous service handling.	
	Note
	This function is empty if 'Use Sync Job Processing' is enabled.
Particularities and Limitations	
<ul style="list-style-type: none">> This function is synchronous.> This function is not reentrant.> This function has to be called by CSM.> This function must not be called by the application.	
Call Context	
<ul style="list-style-type: none">> This function can be called from task level only.	

Table 5-31 Cry_30_Rh850Icum_CmacAes128VerMainFunction

5.4.26 Cry_30_Rh850Icum_KeyExtractStart

Prototype	
void Cry_30_Rh850Icum_KeyExtractStart (Csm_ConfigIdType cfgId)	
Parameter	
cfgPtr	Pointer to ConfigStructure
Return code	
CSM_E_OK	Request successful.
CSM_E_NOT_OK	Request failed.
CSM_E_BUSY	Request failed, service is still busy.
Functional Description	
This interface shall be used to initialize the KeyExtract service.	
Particularities and Limitations	
<ul style="list-style-type: none">> This function can be synchronous or asynchronous.> This function is non-reentrant.> This function is called by application.	
Call Context	
<ul style="list-style-type: none">> This function can be called from task level only.	

Table 5-32 Cry_30_Rh850Icum_KeyExtractStart

5.4.27 Cry_30_Rh850Icum_KeyExtractUpdate

Prototype	
void Cry_30_Rh850Icum_KeyExtractUpdate (Csm_ConfigIdType cfgId, const uint8* dataPtr, uint32 dataLength)	
Parameter	
cfgPtr	Pointer to ConfigStructure
dataPtr	Holds a pointer to the data which contains either <ul style="list-style-type: none">- a plaintext key (Length is 16)- Messages M1, M2, M3 with an optional prepending KeyId to update a keyslot in the SHE. (Length is 64 or 65 Bytes)- A KeyId (Length is 1)
dataLength	Holds the length of the data in bytes.
Return code	
CSM_E_OK	Request successful.
CSM_E_NOT_OK	Request failed.
Functional Description	
This interface shall be used to feed the KeyExtract service with input data.	
Particularities and Limitations	
<ul style="list-style-type: none">> This function can be synchronous or asynchronous.> This function is non-reentrant.> This function is called by application.	
Call Context	
<ul style="list-style-type: none">> This function can be called from task level only.	

Table 5-33 Cry_30_Rh850Icum_KeyExtractUpdate

5.4.28 Cry_30_Rh850Icum_KeyExtractFinish

Prototype	
void Cry_30_Rh850Icum_KeyExtractFinish (Csm_ConfigIdType cfgId, Csm_SymKeyType* keyPtr)	
Parameter	
cfgPtr	Pointer to ConfigStructure
keyPtr	Holds a pointer to a structure where the result (i.e. the symmetrical key) is stored in.
Return code	
CSM_E_OK	Request successful.
CSM_E_NOT_OK	Request failed.
Functional Description	
This interface shall be used to finish KeyExtract.	
Particularities and Limitations	
<ul style="list-style-type: none">> This function can be synchronous or asynchronous.> This function is non-reentrant.> This function is called by application.	
Call Context	
<ul style="list-style-type: none">> This function can be called from task level only.	

Table 5-34 Cry_30_Rh850Icum_KeyExtractFinish

5.4.29 Cry_30_Rh850Icum_KeyExtractMainFunction


Prototype	
void Cry_30_Rh850Icum_KeyExtractMainFunction (void)	
Parameter	
-	
Return code	
-	
Functional Description	
This function implements the asynchronous service handling.	
	Note
	This function is empty if 'Use Sync Job Processing' is enabled.
Particularities and Limitations	
<ul style="list-style-type: none">> This function is synchronous.> This function is not reentrant.> This function has to be called by CSM.> This function must not be called by the application.	
Call Context	
<ul style="list-style-type: none">> This function can be called from task level only.	

Table 5-35 Cry_30_Rh850Icum_KeyExtractMainFunction

5.4.30 Cry_30_Rh850Icum_KeyWrapSymStart

Prototype	
<pre>void Cry_30_Rh850Icum_KeyWrapSymStart (Csm_ConfigIdType cfgId, const Csm_SymKeyType * keyPtr, const Csm_SymKeyType * wrappingKeyPtr)</pre>	
Parameter	
cfgPtr	Pointer to ConfigStructure
keyPtr	Holds a pointer to the symmetric key to be wrapped.
wrappingKeyPtr	Holds a pointer to the key used for wrapping.
Return code	
CSM_E_OK	Request successful.
CSM_E_NOT_OK	Request failed.
CSM_E_BUSY	Request failed, service is still busy.
Functional Description	
This interface shall be used to initialize the symmetrical key wrapping.	
Particularities and Limitations	
<ul style="list-style-type: none">> This function can be synchronous or asynchronous.> This function is non-reentrant.> This function is called by application.	
Call Context	
<ul style="list-style-type: none">> This function can be called from task level only.	

Table 5-36 Cry_30_Rh850Icum_KeyWrapSymStart

5.4.31 Cry_30_Rh850Icum_KeyWrapSymUpdate

Prototype	
<code>void Cry_30_Rh850Icum_KeyWrapSymUpdate (Csm_ConfigIdType cfgId, const uint8* dataPtr, uint32 * dataLengthPtr)</code>	
Parameter	
cfgPtr	Pointer to ConfigStructure
dataPtr	Holds a pointer to the memory location which will hold the result of the key wrapping.
dataLengthPtr	Holds a pointer to the memory location in which the length information is stored. On calling this function this parameter shall contain the size of the buffer provided by dataPtr. When the request has finished, the actual length of the computed value shall be stored. Valid value is 112.
Return code	
CSM_E_OK	Request successful.
CSM_E_NOT_OK	Request failed.
Functional Description	
This interface shall be used to retrieve the result of the key wrapping operation from the symmetrical key wrapping.	
Particularities and Limitations	
<ul style="list-style-type: none">> This function can be synchronous or asynchronous.> This function is non-reentrant.> This function is called by application.	
Call Context	
<ul style="list-style-type: none">> This function can be called from task level only.	


Table 5-37 Cry_30_Rh850Icum_KeyWrapSymUpdate

5.4.32 Cry_30_Rh850Icum_KeyWrapSymFinish

Prototype	
void Cry_30_Rh850Icum_KeyWrapSymFinish (Csm_ConfigIdType cfgId)	
Parameter	
cfgPtr	Pointer to Config Structure
Return code	
CSM_E_OK	Request successful.
CSM_E_NOT_OK	Request failed.
Functional Description	
This interface shall be used to finish the symmetrical key wrapping.	
Particularities and Limitations	
<ul style="list-style-type: none">> This function can be synchronous or asynchronous.> This function is non-reentrant.> This function is called by application.	
Call Context	
<ul style="list-style-type: none">> This function can be called from task level only.	

Table 5-38 Cry_30_Rh850Icum_KeyWrapSymFinish

5.4.33 Cry_30_Rh850Icum_KeyWrapSymMainFunction

Prototype	
void Cry_30_Rh850Icum_KeyWrapSymMainFunction (void)	
Parameter	
-	
Return code	
-	
Functional Description	
This function implements the asynchronous service handling.	
<div>Note This function is empty if 'Use Sync Job Processing' is enabled.</div>	
Particularities and Limitations	
<ul style="list-style-type: none">> This function is synchronous.> This function is not reentrant.> This function has to be called by CSM.> This function must not be called by the application.	

Call Context

> This function can be called from task level only.

Table 5-39 Cry_30_Rh850Icum_KeyWrapSymMainFunction

5.5 Services used by CRY_30_RH850ICUM

In the following table services provided by other components, which are used by the CRY_30_RH850ICUM are listed. For details about prototype and functionality refer to the documentation of the providing component.

Component	API
CSM	Csm_<Service>CallbackNotification Csm_<Service>ServiceFinishNotification
R_ICUMIF	R_ICUMIF_Init R_ICUMIF_ServiceRequest R_ICUMIF_ServiceResponse R_ICUMIF_IsServiceCompleted

Table 5-40 Services used by the CRY_30_RH850ICUM

5.6 Service Ports

The current implementation of the CRY does not support Service Ports.

6 Configuration

In the CRY_30_RH850ICUM the attributes can be configured with the following tools:

- > Configuration in DaVinci Configurator 5

6.1 Configuration Variants

The CRY_30_RH850ICUM supports the configuration variants

- > VARIANT-PRE-COMPIL

6.2 Configuration with DaVinci Configurator 5

6.2.1 Common Properties

Attribute Name	Values ¹	Description
CryUseSyncJobProcessing	STD_ON STD_OFF	Preprocessor switch to enable and disable synchronous job processing.
CryVersionInfoApi	STD_ON STD_OFF	Preprocessor switch to enable and disable availability of the API Cry_GetVersionInfo(). True: API Cry_GetVersionInfo() is available. False: API Cry_GetVersionInfo() is not available.

Table 6-1 Common configuration properties

6.2.2 AES Encrypt Properties

Attribute Name	Values	Description
CryAesEncrypt128BlockMode	CRY_AESBLOCKMODE_CBC CRY_AESBLOCKMODE_ECB	The block mode describes how to handle data which exceeds the block length.
CryKeyIdType	CRY_KEYIDTYPE_RAW CRY_KEYIDTYPE_MAPPED	Defines how a passed keyId is interpreted. Refer to chapter "3.5 Key Mapping" for details.

Table 6-2 Configuration properties of AES-128 Encrypt

6.2.3 AES Decrypt Properties

Attribute Name	Values	Description
CryAesDecrypt128BlockMode	CRY_AESBLOCKMODE_CBC CRY_AESBLOCKMODE_ECB	The block mode describes how to handle data which exceeds the block length.
CryKeyIdType	CRY_KEYIDTYPE_RAW CRY_KEYIDTYPE_MAPPED	Defines how a passed keyId is interpreted. Refer to chapter "3.5 Key Mapping" for details.

Table 6-3 Configuration properties of AES-128 Decrypt

¹ Default values are typed bold

6.2.4 CMAC AES-128 Verification Properties

Attribute Name	Values	Description
CryLengthInBytes	TRUE FALSE	If TRUE, the given mac length is interpreted as the number of bytes to verify. Otherwise the length is interpreted as the number of bits, which are then verified from MSB to LSB. Example: If mac length in bit is 4, the 4 most significant bits are verified. The other 4 less significant bits are discarded.
CryKeyIdType	CRY_KEYIDTYPE_RAW CRY_KEYIDTYPE_MAPPED	Defines how a passed keyId is interpreted. Refer to chapter "3.5 Key Mapping" for details.

Table 6-4 Configuration properties of CMAC AES-128 Verification

6.2.5 CMAC AES-128 Generation Properties

Attribute Name	Values	Description
CryKeyIdType	CRY_KEYIDTYPE_RAW CRY_KEYIDTYPE_MAPPED	Defines how a passed keyId is interpreted. Refer to chapter "3.5 Key Mapping" for details.

Table 6-5 Configuration properties of CMAC AES-128 Generation

6.2.6 Key Extract Properties

Attribute Name	Values	Description
CryKeyIdType	CRY_KEYIDTYPE_RAW CRY_KEYIDTYPE_MAPPED	Defines how a passed keyId is interpreted. Refer to chapter "3.5 Key Mapping" for details.

Table 6-6 Configuration properties of Key Extract

6.2.7 Key Wrap Sym Properties

Attribute Name	Values	Description
CryKeyIdType	CRY_KEYIDTYPE_RAW CRY_KEYIDTYPE_MAPPED	Defines how a passed keyId is interpreted. Refer to chapter "3.5 Key Mapping" for details.

Table 6-7 Configuration properties of Key Wrap Sym

6.3 Deviations

The current implementation does not have any deviations.

6.4 Additions/ Extensions

The current implementation does not have any extensions.

6.5 Limitations

6.5.1 Support of Cryptographic Services

The current cryptographic services are supported:

▶ SHE-AES128-Service for Symmetrical Interface
▶ SHE-PRNG-Service for Random Interface
▶ SHE-CMAC-Service for MAC Interface
▶ Service for Symmetrical Key Extract Interface
▶ Service for Symmetrical Key Wrapping Interface

Table 6-8 Supported AUTOSAR standard conform features

6.5.2 Tool supported configuration

Currently, a tool supported configuration is not implemented. Therefore, the CRY module must be configured manually by editing the configuration files.

6.5.3 Parallel Access to Services

Due to limitations in the SHE, it's not possible to process more than one service at once. An error (CSM_E_BUSY) is generated when a service is already running and another service tries to start at the same time.

Therefore parallel access to services which are dependent on the SHE is not allowed.

7 Glossary and Abbreviations

7.1 Glossary

Term	Description
Cryptographic Primitive	An underlying cryptographic module or library

Table 7-1 Glossary

7.2 Abbreviations

Abbreviation	Description
API	Application Programming Interface
AUTOSAR	Automotive Open System Architecture
BSW	Basis Software
CRY	Cryptographic library module
CSM	Crypto Service Manager
DEM	Diagnostic Event Manager
DET	Development Error Tracer
ECU	Electronic Control Unit
HIS	Hersteller Initiative Software
MICROSAR	Microcontroller Open System Architecture (the Vector AUTOSAR solution)
RTE	Runtime Environment
SchM	Schedule Manager
SHE	Secure Hardware Extension
SRS	Software Requirement Specification
SWC	Software Component
SWS	Software Specification

Table 7-2 Abbreviations

8 Contact

Visit our website for more information on

- > News
- > Products
- > Demo software
- > Support
- > Training data
- > Addresses

www.vector.com