MICROSAR Crypto Abstraction Library

Document Information

History

Author	Date	Version	Remarks
	- -		
	- -		
	- -		
	- -		

Reference Documents

No.	Source	Title	Version
[1]			
[2]			

Scope of the Document

©

Contents

Illustrations

Tables

©

1 Introduction

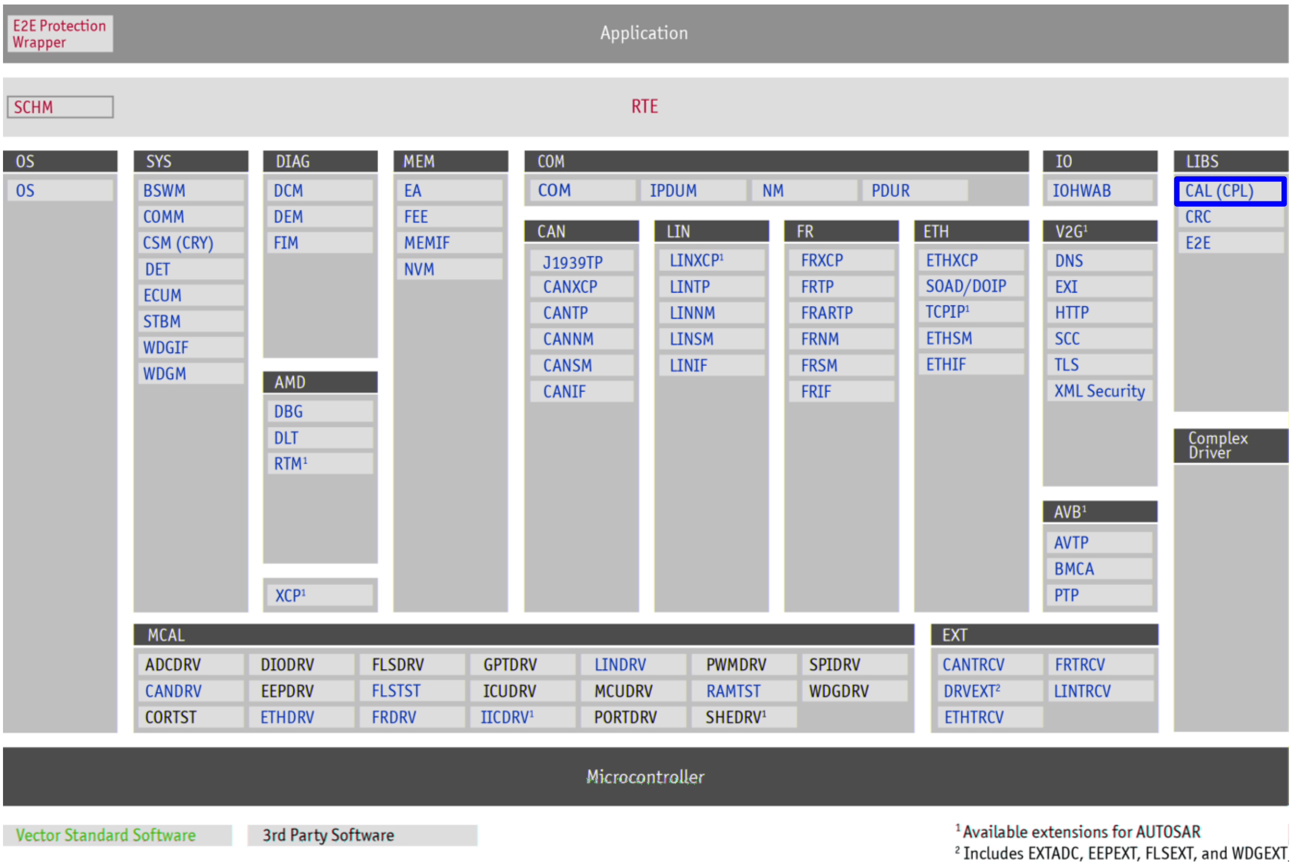
Supported AUTOSAR Release:		
Supported Configuration Variants:	-	
Vendor ID:		-
Module ID:		

- **Symmetrical Decryption/Encryption Interface:**
- **Random Interface:** -
- **Signature Verify Interface:**
- **Hash Interface:**
- **Symmetrical Key Extract Interface:**
- **Symmetrical Block Interface:**

- **MAC Generate and Verify Interface:**

- **Key Derivation Interface:**
- **Key Exchange Interface:**

1.1 Architecture Overview

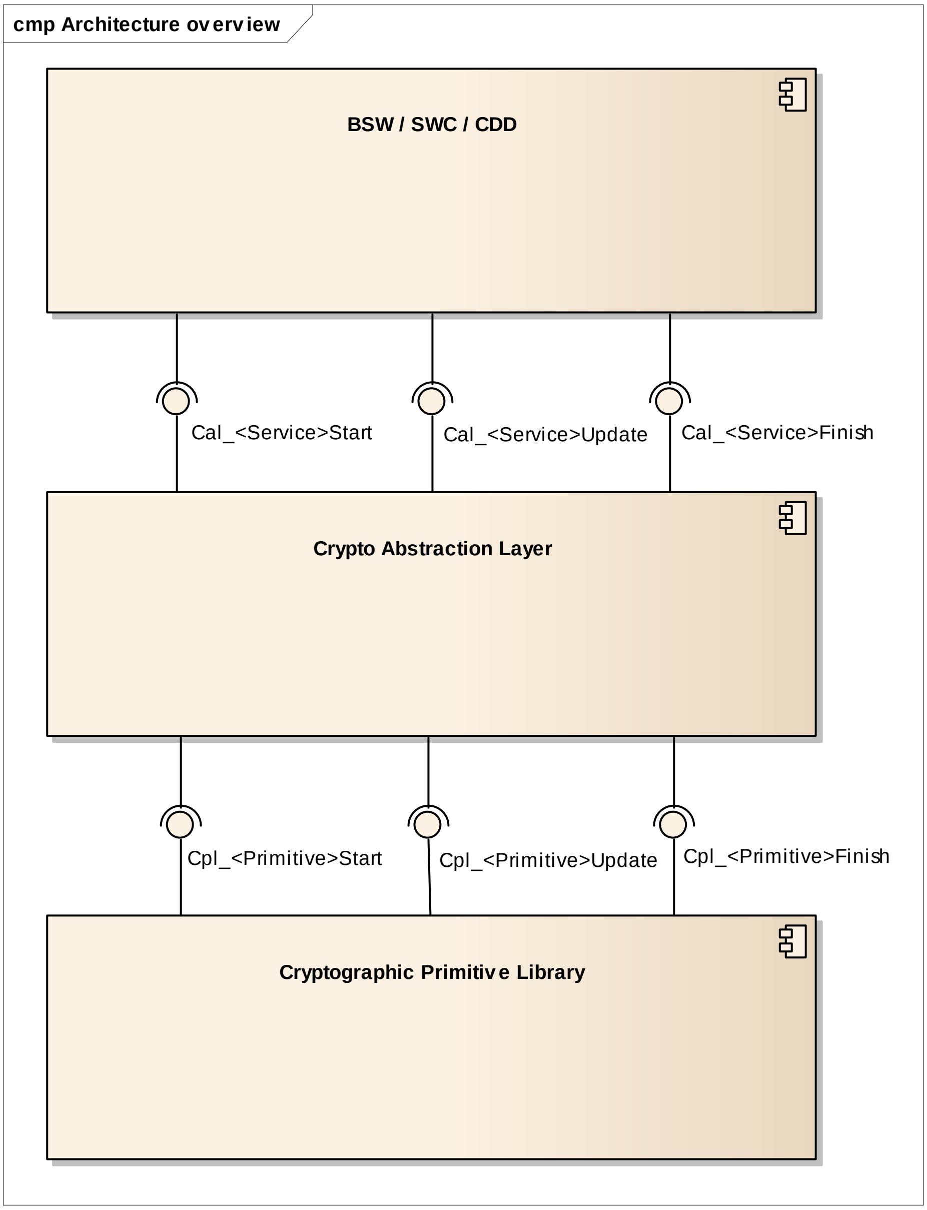


[1] Available extensions for AUTOSAR

[2] Includes EXTADC, EEPEXT, FLSEXT, and WDGEXT

-

©


cmp Architecture overview
BSW / SWC / CDD
Cal_<Service>Start
Cal_<Service>Update
Cal_<Service>Finish
Crypto Abstraction Layer
Cpl_<Primitive>Start
Cpl_<Primitive>Update
Cpl_<Primitive>Finish
Cryptographic Primitive Library

-

2 Functional Description

2.1 Features

▶ -

▶ -

Supported AUTOSAR Standard Conform Features

-

Not Supported AUTOSAR Standard Conform Features

-

Features Provided Beyond The AUTOSAR Standard

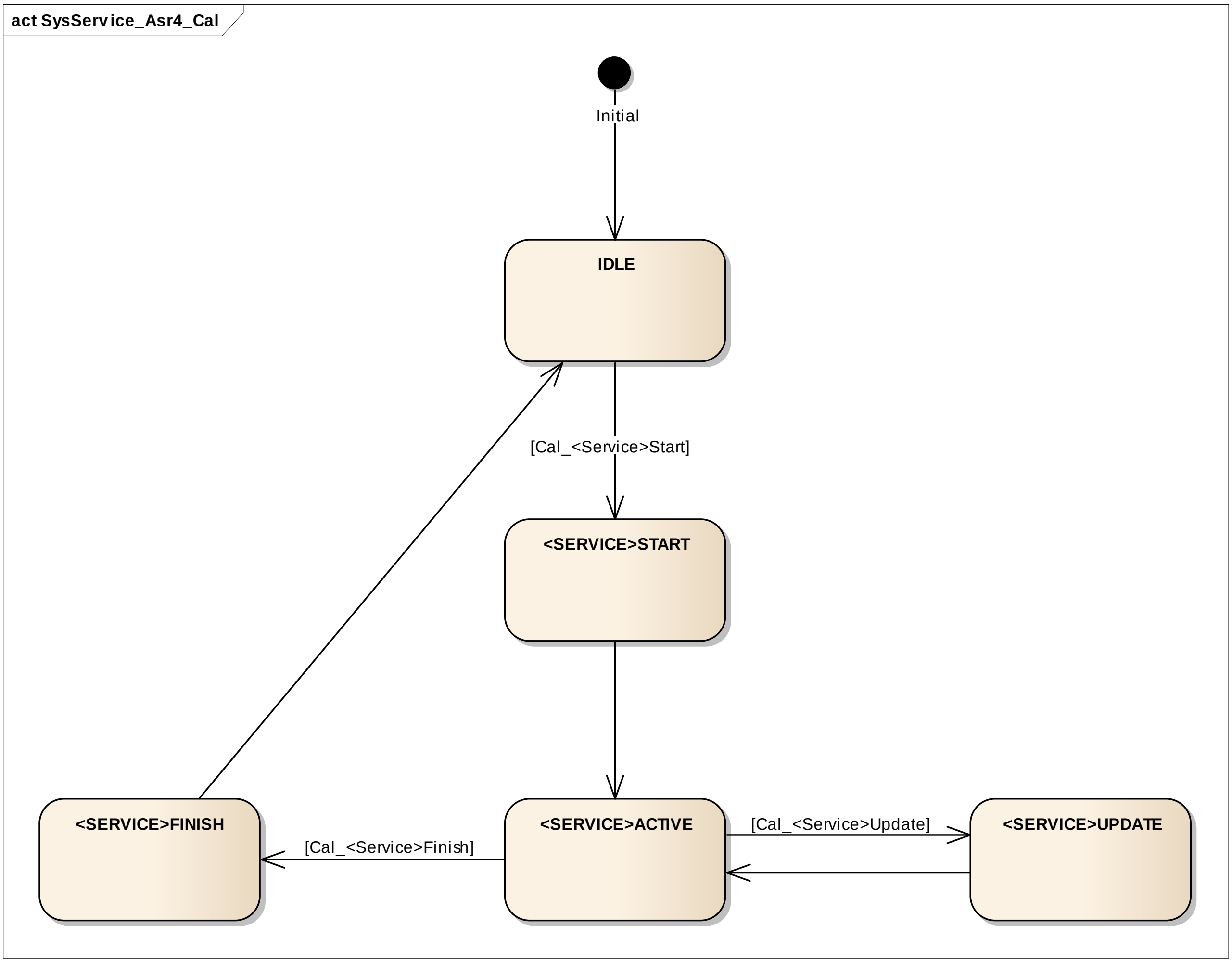
-

©

2.2 Initialization

2.3 States

Cal_<service>Start() **Cal_<service>Update()** **Cal_<service>Finish**



-

2.3.1 Streaming Approach

-

2.4 Error Handling

2.4.1 Development Error Reporting

2.4.2 Production Code Error Reporting

3 Integration

3.1 Scope of Delivery

3.1.1 Static Files

File Name	Source Code Delivery	Object Code Delivery	Description
	■		
	■		
	■		
	■		
	■		
	■		
	■		
	■		
	■		
	■		
	■		
	■		
	■		
	■		
	■		

File Name	Source Code Delivery	Object Code Delivery	Description
[1]		■	

-

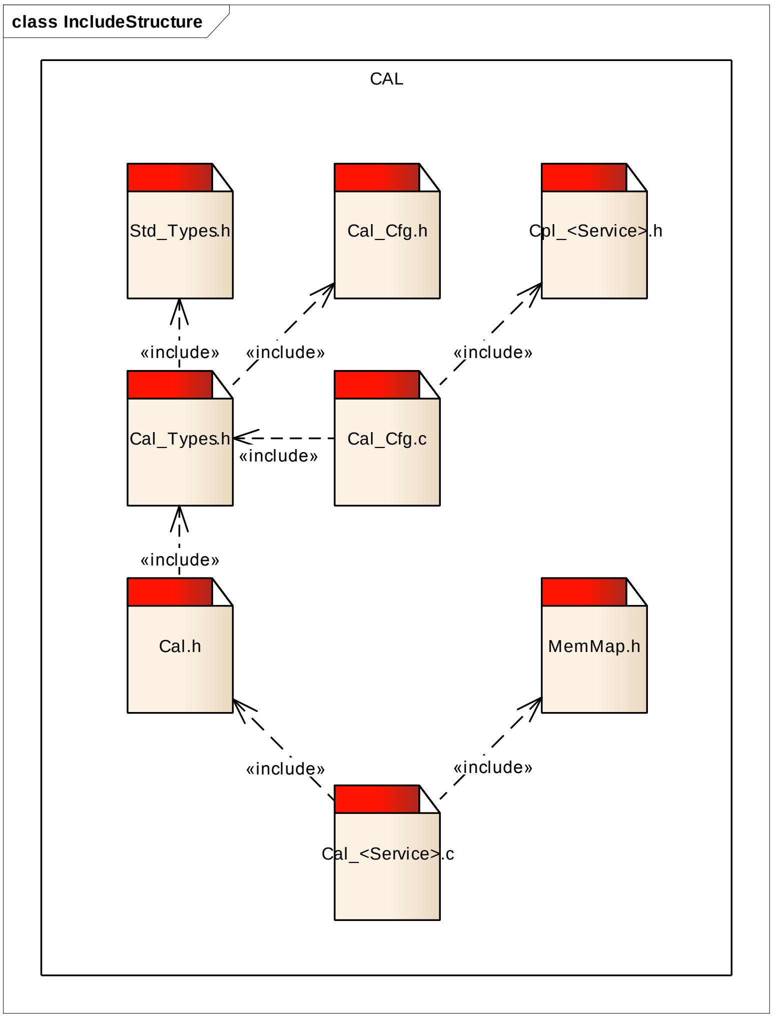
3.1.2 Dynamic Files

File Name	Description

-

[1]

3.1.3 Include Structure



-

3.2 Compiler Abstraction and Memory Mapping

Compiler Abstraction Definitions Memory Mapping Sections		
		■
	■	

-

©

4 API Description

-

4.1 Type Definitions

Type Name	C-Type	Description	Value Range
			CAL_E_OK CAL_E_NOT_OK CAL_E_SMALL_BUFFER CAL_E_ENTROPY_EXHAUSTION
			CAL_ACT_IDLE CAL_ACT_ACTIVE

-

Cal_<Service>ConfigType

Struct Element Name	C-Type	Description	Value Range

-

Cal_<Primitive>ConfigType

[redacted]

Struct Element Name	C-Type	Description	Value Range

©

Struct Element Name	C-Type	Description	Value Range

-

Cal_<Service>CtxBufType[]

4.2 Services provided by CAL

4.2.1 Cal_SymDecryptStart

Prototype	
`Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_SymDecryptCtxBufType contextBuffer, const Cal_SymKeyType *keyPtr, const uint8 *InitVectorPtr, uint32 InitVectorLength)`	
Parameter	
Return code	
	- -
Functional Description	
Particularities and Limitations	

-

4.2.2 Cal_SymDecryptUpdate

Prototype	
`Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_SymDecryptCtxBufType contextBuffer, const uint8 *cipherTextPtr, uint32 cipherTextLength, uint8 *plainTextPtr, uint32 *plainTextLengthPtr)`	
Parameter	
Return code	
	- - -
Functional Description	
Particularities and Limitations	

-

4.2.3 Cal_SymDecryptFinish

Prototype	
`Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_SymDecryptCtxBufType contextBuffer, uint8 *plainTextPtr, uint32 *plainTextLengthPtr)`	
Parameter	

©

©

Particularities and Limitations

-

4.2.5 Cal_SymEncryptUpdate

Prototype	
Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_SymDecryptCtxBufType contextBuffer, const uint8 *plainTextPtr, uint32 plainTextLength, uint8 *cipherTextPtr, uint32 *cipherTextLengthPtr)	
Parameter	
Return code	
	- - -
Functional Description	
Particularities and Limitations	

-

©

4.2.6 Cal_SymEncryptFinish

Prototype	
Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_SymDecryptCtxBufType contextBuffer, uint8 *cipherTextPtr, uint32 *cipherTextLengthPtr)	
Parameter	
Return code	
	- - -
Functional Description	
Particularities and Limitations	

-

4.2.7 Cal_KeyDeriveStart

Prototype	
Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_KeyDeriveCtxBufType contextBuffer, uint32 keyLength, uint32 iterations)	
Parameter	

Return code	
	- -
Functional Description	
Particularities and Limitations	

-

4.2.8 Cal_KeyDeriveUpdate

Prototype	
`Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_KeyDeriveCtxBufType contextBuffer, const uint8 *passwordPtr, uint32 passwordLength, const uint8 *saltPtr, uint32 saltLength)`	
Parameter	
Return code	
	- -
Functional Description	
Particularities and Limitations	

-

4.2.9 Cal_KeyDeriveFinish

Prototype	
Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_KeyDeriveCtxBufType contextBuffer, uint8 *keyPtr)	
Parameter	
Return code	
	- -
Functional Description	
Particularities and Limitations	

-

4.2.10 Cal_KeyExchangeCalcSecretStart

Prototype	
Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_KeyExchangeCalcSecretCtxBufType contextBuffer, const Cal_KeyExchangeBaseType basePtr, const Cal_KeyExchangePrivateType privateValuePtr)	
Parameter	

Return code	
	- -
Functional Description	
Particularities and Limitations	

-

4.2.11 Cal_KeyExchangeCalcSecretUpdate

Prototype	
`Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_KeyExchangeCalcSecretCtxBufType contextBuffer, const uint8 *partnerPublicValuePtr, uint32 partnerPublicValueLength)`	
Parameter	
Return code	
	- -
Functional Description	
Particularities and Limitations	

-

©

Return code	
	- -
Functional Description	
Particularities and Limitations	

-

4.2.14 Cal_RandomSeedStart

Prototype	
`Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_RandomCtxBufType contextBuffer)`	
Parameter	
Return code	
	- -
Functional Description	
Particularities and Limitations	

-

4.2.15 Cal_RandomSeedUpdate

Prototype	
Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_RandomCtxBufType contextBuffer, const uint8 *seedPtr, uint32 seedLength)	
Parameter	
Return code	
	- -
Functional Description	
Particularities and Limitations	

-

4.2.16 Cal_RandomSeedFinish

Prototype	
Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_RandomCtxBufType contextBuffer)	
Parameter	
Return code	
	- -
Functional Description	
Particularities and Limitations	

-

4.2.17 Cal_RandomGenerate

Prototype	
`Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_RandomCtxBufType contextBuffer, uint8 *resultPtr, uint32 resultLength)`	
Parameter	
Return code	
	- - -
Functional Description	
Particularities and Limitations	

-

4.2.18 Cal_SignatureVerifyStart

Prototype
`Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_SignatureVerifyCtxBufType contextBuffer, const Cal_AsymPublicKeyType *keyPtr)`

Parameter	
Return code	
	- -
Functional Description	
Particularities and Limitations	

-

4.2.19 Cal_SignatureVerifyUpdate

Prototype	
`Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_SignatureVerifyCtxBufType contextBuffer, const uint8 *dataPtr, uint32 dataLength)`	
Parameter	
Return code	
	- -
Functional Description	
Particularities and Limitations	

-

4.2.20 Cal_SignatureVerifyFinish

Prototype	
`Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_SignatureVerifyCtxBufType contextBuffer, const uint8 *signaturePtr, uint32 signatureLength, Cal_VerifyResultType *resultPtr)`	
Parameter	
Return code	
	- -
Functional Description	
Particularities and Limitations	

-

4.2.21 Cal_HashStart

Prototype	
`Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_HashCtxBufType contextBuffer)`	
Parameter	
Return code	
	- -

©

Functional Description
Particularities and Limitations

-

4.2.22 Cal_HashUpdate

Prototype	
`Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_HashCtxBufType contextBuffer, const uint8 *dataPtr, uint32 dataLength)`	
Parameter	
Return code	
	- -
Functional Description	
Particularities and Limitations	

-

4.2.23 Cal_HashFinish

Prototype
`Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_HashCtxBufType contextBuffer, uint8 *resultPtr, uint32 *resultLengthPtr, boolean truncationIsAllowed)`

Parameter	
Return code	
	- - -
Functional Description	
Particularities and Limitations	

-

4.2.24 Cal_SymKeyExtractStart

Prototype	
`Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_SymKeyExtractCtxBufType contextBuffer)`	
Parameter	
Return code	
	- -
Functional Description	

©

Particularities and Limitations

-

4.2.25 Cal_SymKeyExtractUpdate

Prototype
`Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_SymKeyExtractCtxBufType contextBuffer, const uint8 *dataPtr, uint32 dataLength)`

Parameter	
	-

Return code	
	- -

Functional Description

Particularities and Limitations

-

4.2.26 Cal_SymKeyExtractFinish

Prototype
`Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_SymKeyExtractCtxBufType contextBuffer, Cal_SymKeyType *keyPtr)`

Parameter	

©

Return code	
	- -
Functional Description	
Particularities and Limitations	

-

4.2.27 Cal_SymBlockEncryptStart

Prototype	
`Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_SymBlockEncryptCtxBufType contextBuffer, const Cal_SymKeyType *keyPtr)`	
Parameter	
Return code	
	- -
Functional Description	
Particularities and Limitations	

-

4.2.28 Cal_SymBlockEncryptUpdate

Prototype	
Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_SymBlockEncryptCtxBufType contextBuffer, const uint8 *plainTextPtr, uint32 plainTextLength, uint8 *cipherTextPtr, uint32 *cipherTextLengthPtr)	
Parameter	
Return code	
	- -
Functional Description	
Particularities and Limitations	

-

4.2.29 Cal_SymBlockEncryptFinish

Prototype	
Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_SymBlockEncryptCtxBufType contextBuffer)	
Parameter	

Return code	
	- -
Functional Description	
Particularities and Limitations	

-

4.2.30 Cal_SymBlockDecryptStart

Prototype	

`Cal_ReturnType (Cal_ConfigIdType cfgId,`
`Cal_SymBlockDecryptCtxBufType contextBuffer, const Cal_SymKeyType`
`*keyPtr)`	

Parameter	
Return code	
	- -
Functional Description	
Particularities and Limitations	

-

4.2.31 Cal_SymBlockDecryptUpdate

Prototype	
`Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_SymBlockDecryptCtxBufType contextBuffer, const uint8 *cipherTextPtr, uint32 cipherTextLength, uint8 *plainTextPtr, uint32 *plainTextLengthPtr)`	
Parameter	
Return code	
	- -
Functional Description	
Particularities and Limitations	

-

4.2.32 Cal_SymBlockDecryptFinish

Prototype	
`Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_SymBlockDecryptCtxBufType contextBuffer)`	
Parameter	

©

Return code	
	- -
Functional Description	
Particularities and Limitations	

-

4.2.33 Cal_MacGenerateStart

Prototype	
`Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_MacGenerateCtxBufType contextBuffer, const Cal_SymKeyType *keyPtr)`	
Parameter	
Return code	
	- -
Functional Description	
Particularities and Limitations	

-

4.2.34 Cal_MacGenerateUpdate

Prototype
`Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_MacGenerateCtxBufType contextBuffer, const uint8 *dataPtr, uint32 dataLength)`

Parameter	

Return code	
	- -

Functional Description

Particularities and Limitations

-

4.2.35 Cal_MacGenerateFinish

Prototype
Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_MacGenerateCtxBufType contextBuffer, uint8 *resultPtr, uint32 *resultLengthPtr, boolean TruncationIsAllowed)

Parameter	

Return code	
	- - -

Functional Description	
Particularities and Limitations	

-

4.2.36 Cal_MacVerifyStart

Prototype	
`Cal_ReturnType                    (Cal_ConfigIdType cfgId, Cal_MacVerifyCtxBufType contextBuffer, const Cal_SymKeyType *keyPtr)`	
Parameter	
Return code	
	- -
Functional Description	
Particularities and Limitations	

-

4.2.37 Cal_MacVerifyUpdate

Prototype	
`Cal_ReturnType                    (Cal_ConfigIdType cfgId, Cal_MacVerifyCtxBufType contextBuffer, const uint8 *dataPtr, uint32 dataLength)`	
Parameter	

Return code	
	- -
Functional Description	
Particularities and Limitations	

-

4.2.38 Cal_MacVerifyFinish

Prototype	
`Cal_ReturnType (Cal_ConfigIdType cfgId, Cal_MacVerifyCtxBufType contextBuffer, const uint8 *MacPtr, uint32 MacLength, Cal_VerifyResultType *resultPtr)`	
Parameter	
Return code	
	- -
Functional Description	
Particularities and Limitations	

-

©

4.3 API used by CAL

Component	API

-

4.4 Callback Functions

4.5 Configurable Interfaces

4.5.1 Notifications

4.5.2 Callout Functions

4.5.3 Hook Functions

©

5 Configuration

-

5.1 Configuration Variants

- VARIANT-PRE-COMPILE

5.2 Configuration with DaVinci Configurator

5.2.1 Common Properties

Attribute Name	Values	Description
	STD_ON	
	STD_ON	
	32	

-

5.2.2 Service Type related Properties

Attribute Name	Values	Description
		This is the maximum size over all key lengths used in all CPL primitives, which implement an <ServiceType> Service. Please note that the calling application has to provide the key buffer. So, it has to be ensured that the size of this buffer matches with the configured value here.
		This is the maximum size over all context buffers used in all CPL primitives, which implement an <ServiceType> Service. Please note that the calling application has to provide the context buffer. So, it has to be ensured that the size of this buffer matches with the configured value here.

-

©

©

6 AUTOSAR Standard Compliance

- -

6.1 Deviation from the AUTOSAR Software Specification

7 Glossary and Abbreviations

7.1 Abbreviations

Abbreviation	Description

-

©

8 Contact