Document Title	Requirements on Crypto Service Manager
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	426
Document Classification	Auxiliary

Document Version	1.0.0
Document Status	Final
Part of Release	4.0
Revision	1

Document Change History

Date	Version	Changed by	Change Description
30.11.2009	1.0.0	AUTOSAR Administration	Initial release

Disclaimer

Advice for users

AUTOSAR specifications may contain exemplary items (exemplary reference models, "use cases", and/or references to exemplary technical solutions, devices, processes or software).

Any such exemplary items are contained in the specifications for illustration purposes only, and they themselves are not part of the AUTOSAR Standard. Neither their presence in such specifications, nor any later documentation of AUTOSAR conformance of products actually implementing such exemplary items, imply that intellectual property rights covering such exemplary items are licensed under the same rules as applicable to the AUTOSAR Standard.

Table of Contents

1 Scope of Document

This document specifies the requirements of the module Crypto Service Manager (CSM).

The integration of the HIS crypto functionality was a planned feature for AUTOSAR release 4.0. For details please refer to feature BRF00165 in the following document:

Requirements on BSW & RTE Features
AUTOSAR_RS_BSWAndRTEFeatures.pdf

The introduction of the CSM module in AUTOSAR realizes this feature.

2 Conventions to be used

- In requirements, the following specific semantics shall be used (based on the Internet Engineering Task Force IETF).

  The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as:

  - SHALL: This word means that the definition is an absolute requirement of the specification.
  - SHALL NOT: This phrase means that the definition is an absolute prohibition of the specification.
  - MUST: This word means that the definition is an absolute requirement of the specification due to legal issues.
  - MUST NOT: This phrase means that the definition is an absolute prohibition of the specification due to legal constraints.
  - SHOULD: This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
  - SHOULD NOT: This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
  - MAY: This word, or the adjective „OPTIONAL“, means that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation, which does not include a particular option, MUST be prepared to interoperate with another implementation, which does include the option, though perhaps with reduced functionality. In the same vein an implementation, which does include a particular option, MUST be prepared to interoperate with another implementation, which does not include the option (except, of course, for the feature the option provides.)

3 Acronyms and abbreviations

Abbreviation / Acronym:	***Description:***
DEM	Diagnostic Event Manager
DET	Development Error Tracer
CSM	Crypto Service Manager
CRY	Cryptographic library module

4 Requirements Specification

4.1 Functional Overview

The Crypto Service Manager (CSM) offers a standardized access to cryptographic services for applications and system functions.

The cryptographic services are, e.g., the computation of hashes, the verification of asymmetrical signatures, or the symmetrical encryption of data. These services depend on underlying cryptographic primitives and cryptographic schemes. The CSM shall make it possible for different applications to use the same service but using different underlying primitives and/or schemes. E.g., one application might need to use the hash service to compute an MD5 digest and another might need to compute an SHA1 digest. Or one application might need to verify a signature which has been computed with the RSASSA-PKCS1-V1_5 signature scheme and using SHA1 as an underlying hash primitive, while another application might need to verify a signature computed with a different scheme which uses MD5 as an underlying hash primitive. The CSM shall make it possible to configure which services are needed and to create several configurations for each service where schemes and primitives can be chosen.

Furthermore, since the computation of many of the cryptographic services is very computation intensive, provisions have to be made for scheduling these long computations. The CSM shall be configurable to use an asynchronous interface where the service requests are placed at the CSM by synchronous interface functions and the services are processed in a main function. Since there is the possibility that there is no operating system which is able to schedule the main function, it should be possible to make the main function interruptible.

To serve the given methods, cryptographic algorithms must be provided by a crypto library. The internal interface to the cryptographic algorithms, named CRY, is defined in a generic and configurable way.

4.2 Functional Requirements

4.2.1 General

4.2.1.1 [BSW42600061] General interfaces

Initiator:	WP Libraries | BMW
Date:	2008-07-01
Short Description:	-
Type:	new
Importance:	high
Description:	The specification of the CSM shall take two possibilities into consideration. If interruption is needed for the intended application, the CSM shall provide a clean interruption interface using an asynchronous interface and an interruptible main function. If no interruption is needed the CSM has also to provide a clean interface, but it is not necessary to define a main function.
Rationale:	-
Use Case:	-
Dependencies:	-
Conflicts:	-
Supporting Material:	-

4.2.1.2 [BSW42600001] scalability

Initiator:	WP Libraries | BMW
Date:	2008-07-01
Short Description:	-
Type:	new
Importance:	high
Description:	The CSM shall guarantee that the unused cryptographic primitives of the underlying crypto library are not compiled into the binary.
Rationale:	-
Use Case:	-
Dependencies:	-
Conflicts:	-
Supporting Material:	-

4.2.1.3 [BSW42600002] CRY interface

Initiator:	WP Libraries | BMW
Date:	2008-07-01
Short Description:	-
Type:	new
Importance:	high
Description:	The CSM shall be able to incorporate modules of the crypto library which have been implemented according to the crypto library requirement specification. This internal CSM API interface is named CRY.

Rationale:	-
Use Case:	-
Dependencies:	-
Conflicts:	-
Supporting Material:	-

4.2.1.4 [BSW42600069] CRY interface specification

Initiator:	WP Libraries | BMW
Date:	2008-10-17
Short Description:	-
Type:	new
Importance:	high
Description:	The CSM shall specify required interfaces to the CRY module.
Rationale:	-
Use Case:	-
Dependencies:	-
Conflicts:	-
Supporting Material:	-

4.2.1.5 [BSW42600010] role of cryptographic primitives

Initiator:	WP Libraries | BMW
Date:	2008-07-01
Short Description:	-
Type:	new
Importance:	high
Description:	The CSM module provides a synchronous and an asynchronous interface for using cryptographic services. To serve the services of the underlying crypto library CRY has to supply implementations of cryptographic primitives.
Rationale:	-
Use Case:	-
Dependencies:	-
Conflicts:	-
Supporting Material:	-

4.2.1.6 [BSW42600011] internal CRY interface

Initiator:	WP Libraries | BMW
Date:	2008-07-01
Short Description:	-
Type:	new
Importance:	high
Description:	Each primitive of the CRY shall belong to exactly one service of the CSM.
Rationale:	-
Use Case:	-
Dependencies:	-
Conflicts:	-
Supporting Material:	-

4.2.2 Configuration

4.2.2.1 [BSW42600004] configuration rules

Initiator:	WP Libraries | BMW
Date:	2008-07-01
Short Description:	-
Type:	new
Importance:	high
Description:	The CSM shall provide configuration rules and constraints to enable plausibility checks of configuration during ECU configuration time where possible.
Rationale:	-
Use Case:	-
Dependencies:	-
Conflicts:	-
Supporting Material:	-

4.2.2.2 [BSW42600005] job processing mode

Initiator:	WP Libraries | BMW
Date:	2008-07-01
Short Description:	-
Type:	new
Importance:	high
Description:	The job processing mode (synchronous or asynchronous) of the CSM shall be defined by statical configuration.
Rationale:	-
Use Case:	-
Dependencies:	-
Conflicts:	-
Supporting Material:	-

4.2.2.3 [BSW42600006] cryptographic services

Initiator:	WP Libraries | BMW
Date:	2008-07-01
Short Description:	-
Type:	new
Importance:	high
Description:	The set of cryptographic services provided by the CSM shall be defined by statical configuration.
Rationale:	-
Use Case:	-
Dependencies:	-
Conflicts:	-
Supporting Material:	-

4.2.2.4 [BSW42600007] other modules

Initiator:	WP Libraries | BMW
Date:	2008-07-01
Short Description:	-
Type:	new
Importance:	high
Description:	The CSM module specification shall specify which other modules are required.
Rationale:	-
Use Case:	-
Dependencies:	-
Conflicts:	-
Supporting Material:	-

4.2.2.5 [BSW42600008] callback function

Initiator:	WP Libraries | BMW
Date:	2008-07-01
Short Description:	-
Type:	new
Importance:	high
Description:	The CSM module specification shall specify how the callback function has to be implemented, if the asynchronous job processing mode is selected.
Rationale:	-
Use Case:	-
Dependencies:	-
Conflicts:	-
Supporting Material:	-

4.2.3 Initialisation

4.2.3.1 [BSW42600009] initialization function

Initiator:	WP Libraries | BMW
Date:	2008-07-01
Short Description:	Cal_Init()
Type:	new
Importance:	high
Description:	The initialization of the CSM module should be done in a separate initialization function. This function shall be named CSM_Init().
Rationale:	-
Use Case:	-
Dependencies:	-
Conflicts:	-
Supporting Material:	-

4.2.3.2 Normal Operation

4.2.3.3 [BSW42600030] streaming approach

Initiator:	WP Libraries | BMW
Date:	2008-07-01
Short Description:	-
Type:	new
Importance:	high
Description:	The CSM module shall use the streaming approach for most provided services (see Software Specification of CSM), i.e. it shall be possible to hand over the input data in small chunks to the service. Therefore these services have to provide the following: · A start function, which is called once and will initialize the service. · An update function, which can be called several times after the start function has been called and which provides the input data in arbitrary chunks to the service. · A finish function, which is called after the complete input data has been given with the update function and which will return the result of the service.
Rationale:	-
Use Case:	-
Depenencies:	-
Conflicts:	-
Supporting Material:	

4.2.3.4 [BSW42600063] streaming approach

Initiator:	WP Libraries | BMW
Date:	2008-07-01
Short Description:	-
Type:	new
Importance:	high
Description:	The implementation of the cryptographic primitives shall be based on the streaming approach with start, update and finish functions when the corresponding interface uses such approach.
Rationale:	-
Use Case:	-
Depenencies:	-
Conflicts:	-
Supporting Material:	-

4.2.4 Fault Operation

4.2.4.1 [BSW42600012] error types

Initiator:	WP Libraries | BMW
Date:	2008-07-01
Short Description:	-
Type:	new

Importance:	high
Description:	The CSM module shall distinguish between the following two types or errors: - errors that can only occur during development - errors that are expected to occur also in production code
Rationale:	-
Use Case:	-
Dependencies:	-
Conflicts:	-
Supporting Material:	-

4.2.4.2 [BSW42600013] development errors

Initiator:	WP Libraries | BMW
Date:	2008-07-01
Short Description:	-
Type:	new
Importance:	high
Description:	The CSM module shall report detected development errors to the Development Error Tracer (DET). The detection and reporting shall be statically configurable with one single preprocessor switch.
Rationale:	-
Use Case:	-
Dependencies:	-
Conflicts:	-
Supporting Material:	-

4.2.4.3 [BSW42600014] development error codes via API

Initiator:	WP Libraries | BMW
Date:	2008-07-01
Short Description:	-
Type:	new
Importance:	high
Description:	The CSM module shall not return specific development error codes via the API. In case of a detected development error the error shall only be reported to the DET. If the API function which detected the error has the return type CSM_ReturnType, it shall return CSM_E_NOT_OK.
Rationale:	-
Use Case:	-
Dependencies:	-
Conflicts:	-
Supporting Material:	-

4.2.4.4 [BSW42600015] parameter checking

Initiator:	WP Libraries | BMW
Date:	2008-07-01
Short Description:	-
Type:	new
Importance:	high
Description:	The CSM shall check passed API parameters for validity. This checking shall

	be statically configurable for those errors that only can occur during development.
Rationale:	-
Use Case:	-
Dependencies:	-
Conflicts:	-
Supporting Material:	-

4.3 Non-Functional Requirements (Qualities)

4.3.1 Software architecture requirement

4.3.1.1 [BSW42600047] abstraction layer

Initiator:	WP Libraries | BMW
Date:	2008-07-01
Short Description:	-
Type:	new
Importance:	high
Description:	The CSM module shall provide an abstraction layer which offers a standardized interface to higher software layers to access cryptographic algorithms.
Rationale:	-
Use Case:	-
Dependencies:	-
Conflicts:	-
Supporting Material:	-

4.3.1.2 [BSW42600064] location in service layer

Initiator:	WP Libraries | BMW
Date:	2008-07-01
Short Description:	-
Type:	new
Importance:	high
Description:	The CSM module shall be located in the Autosar service layer
Rationale:	-
Use Case:	-
Dependencies:	-
Conflicts:	-
Supporting Material:	-

4.3.1.3 [BSW42600066] general RTE interface

Initiator:	WP Libraries | BMW
Date:	2008-09-18
Short Description:	-
Type:	new

Importance:	high
Description:	The CSM shall provide an interface to be accessible via the RTE.
Rationale:	-
Use Case:	-
Dependencies:	-
Conflicts:	-
Supporting Material:	-

4.3.1.4 [BSW42600067] RTE interface for services

Initiator:	WP Libraries | BMW
Date:	2008-09-18
Short Description:	-
Type:	new
Importance:	high
Description:	The CSM shall provide one Provide-Port for each configuration. All configured services shall be accessible via this port.
Rationale:	-
Use Case:	-
Dependencies:	-
Conflicts:	-
Supporting Material:	-

4.3.1.5 [BSW42600068] RTE interface for callbacks

Initiator:	WP Libraries | BMW
Date:	2008-09-18
Short Description:	-
Type:	new
Importance:	high
Description:	The CSM shall provide one Require-Port for each configuration. The configured callback function shall be accessible via this port.
Rationale:	-
Use Case:	-
Dependencies:	-
Conflicts:	-
Supporting Material:	-

4.3.2 Software integration requirements

4.3.2.1 [BSW42600060] configuration files

Initiator:	WP Libraries | BMW
Date:	2008-07-01
Short Description:	-
Type:	new
Importance:	high
Description:	The configuration files of the CRY module shall be readable for human beings: e.g. By integration of comments or by tool – support.

Rationale:	-
Use Case:	-
Dependencies:	-
Conflicts:	-
Supporting Material:	-

4.3.3 Software module design requirements

4.3.3.1 [BSW42600036] implementation

Initiator:	WP Libraries | BMW
Date:	2008-07-01
Short Description:	-
Type:	new
Importance:	high
Description:	The implementation shall be conform to MISRA 2004.
Rationale:	-
Use Case:	-
Dependencies:	-
Conflicts:	-
Supporting Material:	-

4.3.3.2 [BSW42600046] error and status information

Initiator:	WP Libraries | BMW
Date:	2008-07-01
Short Description:	-
Type:	new
Importance:	high
Description:	The CRY module shall strictly separate error and status information. This requirement applies to return values and also to internal variables.
Rationale:	-
Use Case:	-
Dependencies:	-
Conflicts:	-
Supporting Material:	-

4.3.3.3 [BSW42600056] files required

Initiator:	WP Libraries | BMW
Date:	2008-07-01
Short Description:	-
Type:	new
Importance:	high
Description:	The CSM module implementation shall provide at least the following files: 1. Module header file: Csm.h 2. Module configuration file: Csm_Cfg.h
Rationale:	-
Use Case:	-

Dependencies:	-
Conflicts:	-
Supporting Material:	-

4.3.3.4 [BSW42600057] configuration and implementation

Initiator:	WP Libraries | BMW
Date:	2008-07-01
Short Description:	-
Type:	new
Importance:	high
Description:	The implementation shall strictly separate the configuration from the implementation.
Rationale:	-
Use Case:	-
Dependencies:	-
Conflicts:	-
Supporting Material:	-