Document Title	Requirements on Cryptography
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	889

Document Status	Final
Part of AUTOSAR Standard	Adaptive Platform
Part of Standard Release	17-10

Document Change History			
Date	**Release**	**Changed by**	**Description**
2017-10-27	17-10	AUTOSAR Release Management	• Initial release

Disclaimer

This work (specification and/or software implementation) and the material contained in it, as released by AUTOSAR, is for the purpose of information only. AUTOSAR and the companies that have contributed to it shall not be liable for any use of the work.

The material contained in this work is protected by copyright and other types of intellectual property rights. The commercial exploitation of the material contained in this work requires a license to such intellectual property rights.

This work may be utilized or reproduced without any modification, in any form or by any means, for informational purposes only. For any other purpose, no part of the work may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The work has been developed for automotive applications only. It has neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Table of Contents

1 Scope of Document

This document specifies requirements on the Crypto Stack of the AUTOSAR Adaptive Platform.

2 Conventions to be used

The representation of requirements in AUTOSAR documents follows the table specified in [TPS_STDT_00078], see Standardization Template [1], chapter Support for Traceability.

The verbal forms for the expression of obligation specified in [TPS_STDT_00053] shall be used to indicate requirements, see Standardization Template [1], chapter Support for Traceability.

3 Requirements Specification

3.1 Functional Overview

The AUTOSAR Adaptive Platform provides functionality to perform cryptographic operations by using standardized interfaces and associated modeling.

3.2 Crypto Stack

[RS_CRYPTO_02001] The Crypto Stack shall conceal symmetric keys from the users ⌈

Type:	draft
Description:	There shall be no interfaces for the users to work with symmetric key values directly. The symmetric key values shall be addressed via identifiers. Symmetric key values shall only be exported in a secure format.
Rationale:	If symmetric key values are available in the application at runtime it increases the risk of key compromise. If symmetric key values are stored in the application, centralized key management (e.g. renewal) is hard.
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋*(RS_Main_00330)*

[RS_CRYPTO_02002] The Crypto Stack shall conceal asymmetric private keys from the users ⌈

Type:	draft
Description:	There shall be no interfaces for the users to work with asymmetric private key values directly. The asymmetric private key values shall be addressed via identifiers. Asymmetric private key values shall only be exported in a secure format.
Rationale:	If asymmetric private key values are available in the application at runtime it increases the risk of key compromise. If asymmetric private key values are stored in the application, centralized key management (e.g. renewal) is hard.
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋*(RS_Main_00330)*

[RS_CRYPTO_02101] The Crypto Stack shall support a primitive to generate cryptographic key material ⌈

Type:	draft

Description:	The Crypto Stack shall support creating cryptographic keys without getting access to the plain key material.
Rationale:	Key confidentiality
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋(*RS_Main_00445*, *RS_Main_00330*, *RS_Main_00514*)

[RS_CRYPTO_02102] The Crypto Stack shall prevent keys from being used in incompatible or insecure ways ⌈

Type:	draft
Description:	The Crypto Stack should detect and prevent use of keys with incompatible algorithms. Keys managed by the Crypto Stack shall be associated with information to detect and prevent use with conflicting or privileged operations.
Dependencies:	–
Use Case:	Protect against unauthorized or incompatible operations that jeapardize confidentiality and integrity of key material (information leakage, key conjuring, API logic attacks).
Supporting Material:	–

⌋(*RS_Main_00514*, *RS_Main_00330*)

[RS_CRYPTO_02103] The Crypto Stack shall support a primitive to derive cryptographic key material from a base key ⌈

Type:	draft
Description:	The Crypto Stack shall support deriving cryptographic keys using a well-defined algorithm from a base key without getting access to the plain key material.
Rationale:	Generating multiple well-defined symmetric keys from a base key
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋(*RS_Main_00445*, *RS_Main_00330*, *RS_Main_00514*)

[RS_CRYPTO_02104] The Crypto Stack shall support a primitive to exchange cryptographic keys with another entity ⌈

Type:	draft
Description:	The Crypto Stack shall support exchanging cryptographic keys without getting access to the plain key material.
Rationale:	Establish common secret
Dependencies:	–
Use Case:	Establish TLS session keys
Supporting Material:	–

⌋(*RS_Main_00445*, *RS_Main_00330*, *RS_Main_00514*)

[RS_CRYPTO_02105] The Crypto Stack shall support a primitive to import and export cryptographic keys into or from a local storage ⌈

Type:	draft
Description:	The Crypto Stack shall support importing and exporting cryptographic keys without getting access to the plain key material.
Rationale:	Support secure distribution of keys from a backend system and/or migration or backup of keys between systems.
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋*(RS_Main_00445, RS_Main_00330, RS_Main_00514, RS_Main_00150)*

[RS_CRYPTO_02201] The Crypto Stack shall provide interfaces to use symmetric encryption and decryption primitives ⌈

Type:	draft
Description:	The Crypto Stack shall support encrypting and decrypting data using an algorithm for symmetric encryption/decryption primitives.
Rationale:	Encrypted communication
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋*(RS_Main_00445, RS_Main_00514, RS_Main_00410)*

[RS_CRYPTO_02202] The Crypto Stack shall provide interfaces to use asymmetric encryption and decryption primitives ⌈

Type:	draft
Description:	The Crypto Stack shall support encrypting and decrypting data using an algorithm for asymmetric encryption/decryption primitives.
Rationale:	Encrypted data
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋*(RS_Main_00445, RS_Main_00514, RS_Main_00410)*

[RS_CRYPTO_02203] The Crypto Stack shall provide interfaces to use message authentication code primitives ⌈

Type:	draft
Description:	The Crypto Stack shall support creating and verifying message authentication codes (MAC).
Rationale:	SecOC using MACs to authenticate messages
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋(*RS_Main_00445*, *RS_Main_00514*, *RS_Main_00410*)

[RS_CRYPTO_02204] The Crypto Stack shall provide interfaces to use digital signature primitives ⌈

Type:	draft
Description:	The Crypto Stack shall support creating and verifying digital signatures.
Rationale:	Digitally signed updates
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋(*RS_Main_00445*, *RS_Main_00514*, *RS_Main_00410*)

[RS_CRYPTO_02205] The Crypto Stack shall provide interfaces to use hashing primitives ⌈

Type:	draft
Description:	The Crypto Stack shall support creating and verifying cryptographic hashes.
Rationale:	Signature verification
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋(*RS_Main_00445*, *RS_Main_00514*, *RS_Main_00410*)

[RS_CRYPTO_02206] The Crypto Stack shall provide interfaces to use random number generation primitives ⌈

Type:	draft
Description:	The Crypto Stack shall support generating cryptographically strong random numbers.
Rationale:	Random numbers are required to generate cryptographic keys, nonces and other inputs to cryptographic protocols.
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋(*RS_Main_00445*, *RS_Main_00514*, *RS_Main_00410*)

[RS_CRYPTO_02207] The Crypto Stack shall provide interfaces to use authenticated symmetric encryption and decryption primitives ⌈

Type:	draft
Description:	The Crypto Stack shall support encrypting and decrypting data using an algorithm for authenticated symmetric encryption/decryption primitives.
Rationale:	Authenticated encrypted communication
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋(*RS_Main_00445*, *RS_Main_00514*, *RS_Main_00410*)

[RS_CRYPTO_02301] The Crypto Stack API shall provide a standardized header files structure ⌈

Type:	draft
Description:	The application shall use standardized header files to abstract from the underlying implementation and platform.
Rationale:	The applications code shall be reusable across different implementations of the AUTOSAR Adaptive platform.
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋(*RS_Main_00060*)

[RS_CRYPTO_02302] The Crypto Stack API shall support a streaming approach ⌈

Type:	draft
Description:	Some primitives are generally used to process large amounts of data. This data may be streamed into the Crypto Stack in multiple smaller pieces.
Rationale:	Basic functionality
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋(*RS_Main_00410*)

[RS_CRYPTO_02303] The Crypto Stack API shall support asynchronous operation ⌈

Type:	draft
Description:	Some primitives may take a long time to complete and thus completion may be communicated in an asynchronous fashion.
Rationale:	Basic functionality
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋(*RS_Main_00410*)

[RS_CRYPTO_02401] The Crypto Stack shall support utilizing multiple concrete implementations of cryptographic algorithms concurrently ⌈

Type:	draft
Description:	Some selected primitives may be implemented in software and/or hardware. Both implementations shall be usable side-by-side in a concurrent fashion.
Rationale:	Basic functionality
Dependencies:	–
Use Case:	–

Supporting Material:	–

⌋(*RS_Main_00445*, *RS_Main_00514*, *RS_Main_00410*)

[RS_CRYPTO_02402] The Crypto Stack shall support prioritizing the processing of requests ⌈

Type:	draft
Description:	Accessing specific functionality can lead to contention that shall be resolved by assigning priorities to requests.
Rationale:	Basic functionality
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋(*RS_Main_00200*)

[RS_CRYPTO_02403] The Crypto Stack shall support isolating keys and requests ⌈

Type:	draft
Description:	In a multi-tenant scenario the Crypto Stack shall implement an individual logical view of available keys and active operations for each tenant.
Dependencies:	–
Use Case:	A application using the Crypto Stack should not be able to observe or manipulate the list of active keys and crypto operations of another application (error injection, timing side-channels, ..).
Supporting Material:	–

⌋(*RS_Main_00514*, *RS_Main_00330*)

[RS_CRYPTO_02404] The Crypto Stack shall support constraining the usage and access to keys and requests ⌈

Type:	draft
Description:	In a multi-tenant scenario the Crypto Stack shall enforce per-tenant usage and access constraints based on security policies and per-tenant access information provided by IAM.
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋(*RS_Main_00514*, *RS_Main_00330*)

4 Requirements Tracing

The following table references the features specified in [2] and links to the fulfillments of these.

Feature	Description	Satisfied by
[RS_Main_00060]	AUTOSAR shall provide a standardized software interface for communication between Applications	[RS_CRYPTO_02301]
[RS_Main_00150]	AUTOSAR shall support the deployment and reallocation of AUTOSAR Application Software	[RS_CRYPTO_02105]
[RS_Main_00200]	AUTOSAR specifications shall allow resource efficient implementations	[RS_CRYPTO_02402]
[RS_Main_00330]	AUTOSAR shall support the principle of information hiding	[RS_CRYPTO_02001] [RS_CRYPTO_02002] [RS_CRYPTO_02101] [RS_CRYPTO_02102] [RS_CRYPTO_02103] [RS_CRYPTO_02104] [RS_CRYPTO_02105] [RS_CRYPTO_02403] [RS_CRYPTO_02404]
[RS_Main_00410]	AUTOSAR shall provide specifications for routines commonly used by Application Software to support sharing and optimization	[RS_CRYPTO_02201] [RS_CRYPTO_02202] [RS_CRYPTO_02203] [RS_CRYPTO_02204] [RS_CRYPTO_02205] [RS_CRYPTO_02206] [RS_CRYPTO_02207] [RS_CRYPTO_02302] [RS_CRYPTO_02303] [RS_CRYPTO_02401]
[RS_Main_00445]	AUTOSAR shall standardize access to crypto-specific HW and SW	[RS_CRYPTO_02101] [RS_CRYPTO_02103] [RS_CRYPTO_02104] [RS_CRYPTO_02105] [RS_CRYPTO_02201] [RS_CRYPTO_02202] [RS_CRYPTO_02203] [RS_CRYPTO_02204] [RS_CRYPTO_02205] [RS_CRYPTO_02206] [RS_CRYPTO_02207] [RS_CRYPTO_02401]

[RS_Main_00514]	AUTOSAR shall support the development of secure systems	[RS_CRYPTO_02101] [RS_CRYPTO_02102] [RS_CRYPTO_02103] [RS_CRYPTO_02104] [RS_CRYPTO_02105] [RS_CRYPTO_02201] [RS_CRYPTO_02202] [RS_CRYPTO_02203] [RS_CRYPTO_02204] [RS_CRYPTO_02205] [RS_CRYPTO_02206] [RS_CRYPTO_02207] [RS_CRYPTO_02401] [RS_CRYPTO_02403] [RS_CRYPTO_02404]

5 References

[1] System Template
AUTOSAR_TPS_SystemTemplate

[2] Requirements on AUTOSAR Features
AUTOSAR_RS_Features