Elektrobit

EB tresos® AutoCore Generic 8 Crypto Driver documentation

release notes update for the Crypto module

product release 8.8.7

Elektrobit Automotive GmbH
Am Wolfsmantel 46
91058 Erlangen, Germany
Phone: +49 9131 7701 0
Fax: +49 9131 7701 6333
Email: info.automotive@elektrobit.com

Technical support

https://www.elektrobit.com/support

Legal disclaimer

Confidential information.

Table of Contents

1. Overview

This document provides you with the release notes to accompany an update to the `Crypto` module. Refer to the changelog Section 2.1, “Change log” for details of changes made for this update.

Release notes details

- EB tresos AutoCore release version: 8.8.7
- EB tresos Studio release version: 29.2.0
- AUTOSAR R4.3 Rev 0
- Build number: B577598

2. Crypto module release notes

- AUTOSAR R4.3 Rev 0
- AUTOSAR SWS document version: 4.3.0
- Module version: 1.7.55.B577598
- Supplier: Elektrobit Automotive GmbH

2.1. Change log

This chapter lists the changes between different versions.

Module version 1.7.55

2022-11-04

- ASCCRYPTO-4152 Fixed known issue: Signature verification fails if CsmSymKeyMaxLength is defined too small and 64 bit support enabled
- Added configuration option "CryptoRandomSeedKeyValidSetEnable" to enable or disable the internal call of Crypto_KeyValidSet for Crypto_RandomSeed API.

Module version 1.7.54

2022-09-16

- Internal module improvement. This module version update does not affect module functionality.

Module version 1.7.53

2022-08-19

- Internal module improvement. This module version update does not affect module functionality.

Module version 1.7.52

2022-07-22

- Changed INIT_BOOLEAN memory sections to INIT_8

Module version 1.7.51

2022-06-10

- Internal module improvement. This module version update does not affect module functionality.

Module version 1.7.50

2022-05-13

- Internal module improvement. This module version update does not affect module functionality.

Module version 1.7.49

2022-04-08

- Internal module improvement. This module version update does not affect module functionality.

Module version 1.7.48

2022-02-18

- Added justifications for MISRA violations.

Module version 1.7.47

2021-10-08

- Implemented HASH primitives SHA3-224, SHA3-256, SHA3-384 and SHA3-512. Added RSA primitives (PSS, PKCS1V15 and OAEP) with HASH primitives SHA3-224, SHA3-256, SHA3-384 and SHA3-512, as the secondary primitive.
- ASCCRYPTO-3517 Fixed known issue: [SW Crypto] AES-CMAC fails if AES-ECB is also configured

Module version 1.7.46

2021-09-17

- Internal module improvement. This module version update does not affect module functionality.

Module version 1.7.45

2021-08-20

- Enhanced the existing check for CPU_TYPE to support 64-bit platforms.
- Added justification for tasking compiler warnings.

Module version 1.7.43

2021-07-28

- Internal module improvement. This module version update does not affect module functionality.

Module version 1.7.42

2021-06-25

- Improved internal handling of CRYPTO_XVIX_XAIX_KE_RANDOM_SEED_COUNT key element in RandomGenerate AES-CTR_DRBG.

Module version 1.7.40

2021-05-28

- Internal module improvement. This module version update does not affect module functionality.

Module version 1.7.38

2021-04-30

- ASCCRYPTO-2875 Fixed known issue: InputLength checked on exclusively requested CRYPTO_OPERATIONMODE_START
- ASCCRYPTO-2832 Fixed known issue: Value of NULL_PTR is assigned to a variable (if the job has the operation mode "START" AND outputLengthPtr == NULL_PTR)

Module version 1.7.36

2021-03-05

- Internal module improvement. This module version update does not affect module functionality.

Module version 1.7.35

2021-01-22

- Internal module improvement. This module version update does not affect module functionality.

Module version 1.7.34

2020-12-18

- ASCCRYPTO-2810 Fixed known issue: RandomGenerate does not report CRYPTO_E_RE_ENTROPY_EXHAUSTED to the DET

Module version 1.7.32

2020-10-23

- ASCCRYPTO-2253 Fixed known issue: Incorrect long number calculations during simultaneous calls of key exchange, key generate, and primitives
- Improved runtime of Hash SHA2.

Module version 1.7.27

2020-07-31

- ASCCRYPTO-1362 Fixed known issue: Add missing handling of persistent key elements
- ASCCRYPTO-1841 Fixed known issue: SignatureGenerate ECC-NIST generates wrong results on Big Endian platforms
- ASCCRYPTO-2375 Fixed known issue: Crypto can generate duplicated defines for a RamBlockDataAddress
- ASCCRYPTO-2241 Fixed known issue: RandomGenerate AES-CTR_DRBG does not return an indication that a reseed is needed

Module version 1.7.26

2020-06-19

- Changed NO_INIT memory sections to CLEARED
- ASCCRYPTO-2237 Fixed known issue: Crypto does not generate correct symbolic names for CryptoKeyElementIds

Module version 1.7.21

2020-02-21

- Internal module improvement. This module version update does not affect module functionality.

Module version 1.7.20

2020-01-24

- ASCCRYPTO-1605 Fixed known issue: AES-CMAC expanded key can lead to alignment problem

Module version 1.7.19

2019-12-06

- Fixed compiler warning when using GHS compiler

Module version 1.7.16

2019-10-11

- ASCCRYPTO-1504 Fixed known issue: Prefix of compiler abstractions is missing the vendorId and vendorApiInfix
- ASCCRYPTO-1566 Fixed known issue: Callable entities syntax does not match the syntax generated by Rte
- ASCCRYPTO-1228 Fixed known issue: Check for skipping main function can lead to memory exception

Module version 1.7.15

2019-08-09

- ASCCRYPTO-1346 Fixed known issue: Crypto does not generate correct symbolic names for CryptoKeys
- ASCCRYPTO-1314 Fixed known issue: Crypto does not generate correct symbolic names for CryptoDriverObjects

- ASCCRYPTO-1500 Fixed known issue: BSW-MODULE-ENTRY-REF-CONDITIONAL entry for Crypto_MainFunction in Crypto_Bswmd.arxml is not generated
- ASCCRYPTO-1498 Fixed known issue: VENDOR-API-INFIX entry in Crypto_Bswmd.arxml is generated incorrectly

Module version 1.7.11

2019-06-14

- ASCCRYPTO-1243 Fixed known issue: Precalculation for MAC keys (AES-CMAC) during start-up may not be done
- ASCCRYPTO-1297 Fixed known issue: AES-CTRDRBG outputs more data than requested
- ASCCRYPTO-1361 Fixed known issue: ECDSA - failure in random number
- ASCCRYPTO-1290 Fixed known issue: Missing exit of exclusive area if CryptoQueueSize is zero
- The ApiInfix (AI) parameter is now part of the Bswmd.
- ASCCRYPTO-1308 Fixed known issue: Vulnerability to side channel attacks due to not secure 'memcmp' implementation
- ASCCRYPTO-1355 Fixed known issue: AES-ECB ENCRYPT service is not functional
- Improved buffersizes for RSA primitives (PSS, PKCS1V15 and OAEP).

Module version 1.7.9

2019-04-12

- Provided internal cipher AES-ECB externally.
- Improved buffer sizes used for RSA based primitives and changed the feature that symmetric, public and private key sizes can be global configured via Csm.
- Improved Elliptic Curve Diffie-Hellman (ECDH) for x25519 and ECC NIST curves secp256r1 and secp384r1.
- Changed values of reference parameters in XDM and BMD file.
- Removed 'myEcuParameterDefinition' from XDM and BMD file.

Module version 1.7.8

2019-03-22

- Implemented Elliptic Curve Diffie-Hellman (ECDH) for ECC NIST curve secp384r1.

Module version 1.7.7

2019-03-15

- Implemented signature verification and generation using RSASSA-PKCS1V15 with SHA1
- Updated key exchange for selection of either x25519 or NIST ECC elliptic curves.
- Implemented Elliptic Curve Diffie-Hellman (ECDH) for ECC NIST curve secp256r1.

Module version 1.7.6

2019-02-15

- ASCCRYPTO-1085 Fixed known issue: EdDSA and ECDSA SignatureVerify return E_NOT_OK if verification was not successful
- ASCCRYPTO-1053 Fixed known issue: Asynchronous processing can lead to undefined behavior
- Implemented SHA1.
- The certificate, key derivation and key exchange interfaces now reject another call of the same function in parallel.

Module version 1.7.5

2018-12-21

- Improved runtime of AES-GCM, AES-CBC, AES-CFB, RSA-RSAES-OAEP, SHA2, AES-CMAC, SHA2-HMAC, RSA-RSASSA-PKCS1-v1.5 and RSA-RSASSA-PSS
- Implemented ECC NIST Secp256r1 signature generation and signature verification, ECDSA.

Module version 1.7.4

2018-11-09

- Created Tresos Error for 64 Bit support enabled
- Created Tresos Error if no CryptoKeys are configured
- Crypto_ProcessJob() now returns CRYPTO_E_JOB_CANCELED when a synchronous job was cancelled during its processing

Module version 1.7.3

2018-09-21

- ASCCRYPTO-839 Fixed known issue: Compile error in AES_CFB decryption
- ASCCRYPTO-838 Fixed known issue: Compile error in RsaSsa-Pkcs1 v1.5 SIGNATUREGENERATE/VERIFY
- ASCCRYPTO-850 Fixed known issue: Compile error in RSAES-OAEP ENCRYPT
- ASCCRYPTO-855 Fixed known issue: Compile error in Crypto_KeyElementGet()

Module version 1.7.2

2018-09-10

- Implemented Det run-time error types
- Implemented AES-CFB encryption and decryption
- ASCCRYPTO-810 Fixed known issue: Compile error if CryptoQueueSize is zero
- Implemented RSAES-OAEP encryption and decryption
- ASCCRYPTO-812 Fixed known issue: Incorrect use of P2CONST macro may produce compiler errors
- ASCCRYPTO-820 Fixed known issue: Wrong outputPtr in asynchronous SINGLECALL AES-CBC DECRYPT

Module version 1.7.1

2018-07-30

- Disabled the NvM configuration if the variant is AUTOSAR
- Added the parameter CryptoInstanceId. This ID is used to discern several crypto drivers in case more than one driver is used in the same ECU.
- ASCCRYPTO-690 Fixed known issue: Memory access violation in the function Crypto_KeyElementCopy()
- ASCCRYPTO-707 Fixed known issue: Crypto_KeyDerive() writes to an index outside the target key element array
- Improved queue implementation
- Adapted check of target size in KeyElementCopy
- ASCCRYPTO-742 Fixed known issue: Crypto_KeyCopy() results in inconsistent target key and returns wrong value
- Adapted handling of key element read/write access
- ASCCRYPTO-174 Fixed known issue: No calculation of random numbers
- Added empty generation modes "verify" and "verify_swcd" in EB Tresos Studio

- Implemented signature verification and generation using RSASSA-PKCS1V15

Module version 1.7.0

2018-06-07

- ASCCRYPTO-666 Fixed known issue: AES-CMAC: Verification is invalid for key sizes larger than 16 bytes in synchronous single call mode
- Disabled NvM dependencies if there are no persistent key elements
- ASCCRYPTO-667 Fixed known issue: AES-CBC: Encryption and decryption fail for input data sizes of less than 16 bytes in asynchronous single call mode
- Implemented signature verification using RSASSA-PSS
- Implemented Certificate Parse and Certificate Verify
- KeyDerive now supports SHA2-256 as pseudorandom primitive

Module version 1.6.1

2018-05-03

- Adapted queuing in a way that a job can be queued only once. This ensures that a job instance is not overwritten by the following service request for the same job instance.
- ASCCRYPTO-628 Fixed known issue: Wrong DET check in AEADDECRYPT leads to aborted job processing
- ASCCRYPTO-629 Fixed known issue: After calling the KeyDerive function, no further processing of Crypto Driver primitives is possible
- ASCCRYPTO-633 Fixed known issue: Asynchronous GCM with mode STREAMSTART or SINGLECALL locks Crypto Driver Object
- ASCCRYPTO-646 Fixed known issue: GCM decryption only works correctly if length of authentication tag is 128 Bits.
- ASCCRYPTO-639 Fixed known issue: GCM only works correctly if key length is 128 bits
- ASCCRYPTO-634 Fixed known issue: Crypto_KeyElementSet() writes to an index outside the key element array
- ASCCRYPTO-654 Fixed known issue: Crypto_KeyDerive() writes to an index outside the key element array
- ASCCRYPTO-653 Fixed known issue: Crypto_KeyElementCopy() fails when the target key element is configured for internal copy only
- ASCCRYPTO-657 Fixed known issue: Calculation of maximum key element size is incorrect

Module version 1.6.0

2018-04-11

- ASCCRYPTO-479 Fixed known issue: Crypto_ECDHKeyExchangeCalcSecret cannot return E_OK
- ASCCRYPTO-473 Fixed known issue: AES-CTRDRBG: Endless loop occurs during synchronous RandomGenerate
- ASCCRYPTO-492 Fixed known issue: Dangling pointers in UPDATE or FINISH in synchronous primitives CBC, SHA, HMAC, SSG or EdDSA
- Updated Crypto_xVIx_xAIx_Bswmd.arxml to handle multiple Crypto Driver Software instances based on Vendor ID and API Infix
- Adapted initialization of keys and key elements during startup: (i) All non-persistent key elements are initialized by their configured init value; (ii) If reading of a persistent key element from NvM failed, and if the key element has not been initialized by NvM, then the key element is initialized by the Crypto Driver; (iii) A key's state is set to valid if none of its persistent key elements has been initialized by the Crypto Driver. If at least one persistent key element was not loaded successfully by the NvM and has been initialized by the Crypto Driver, the corresponding key is set to invalid. (iv) The initialization of key elements now also considers the precalculation of keys for AES CMAC/ECB.
- Adapted storage of persistent key elements to NvM: (i) Crypto_KeyElementSet() no longer calls NvM_-WriteBlock() after setting the key element's value; (ii) Crypto_KeyValidSet() calls NvM_WriteBlock() for all persistent key elements after setting the key's state to valid.
- Improved the handling of Csm_SymKeyType, Csm_AsymPrivateKeyType, and Csm_AsymPublicKeyType in Crypto. If they are configured in Csm, they are just typecasted to internal Crypto types. If they do not exist in Csm, they are created internally in Crypto, using the max size of the key element with ID 1 that is configured in the keys.
- Implemented the key management function KeyDerive (KDF)
- ASCCRYPTO-615 Fixed known issue: GCM handles authentication tag incorrectly if plaintext length is a multiple of 16 bytes

Module version 1.5.2

2018-03-06

- Removed NG Generator workaround caused by a EB tresos Studio framework bug
- ASCCRYPTO-432 Fixed known issue: The Crypto module does not compile if only the AEAD decryption is enabled
- ASCCRYPTO-421 Fixed known issue: Random service with algo mode CTRDRBG fails for asynchronous processing
- Corrected replacement of VendorApiInfix when lowercase letters are used

- Improved the Crypto_xVIx_xAIx_AL_KeyCopy function so that no processing takes place when the source key is invalid
- Improved the Crypto_xVIx_xAIx_AL_KeyElementSet function so that it uses the Crypto_xVIx_xAIx_-KeyElementSet function to set the key element instead of copying it manually
- Added NG Generator workaround caused by a EB tresos Studio framework bug, as required for EB tresos Studio version 14.5.1 up to less than 24.0.0.

Module version 1.5.1

2018-02-16

- Added support for the configuration of the KeyManagement function RandomSeed() via the RANDOM_ALGORITHM key element
- Improved handling of KeyManagement function call for busy Crypto Driver Object
- Improved handling of outputLengthPtr for every primitive

Module version 1.5.0

2018-02-09

- Added support for SHA2-224, SHA2-256, SHA2-384 and SHA2-512 in the same configuration
- ASCCRYPTO-369 Fixed known issue: SIGNATUREGENERATE/VERIFY (EdDSA) does not reset busy pointer in error case
- Implemented the SipHash-2-4 algorithm using 64 bit tag (message digest)
- Added support for multiple configurations of software Crypto Drivers
- Improved run-time of EdDSA generate and verify
- Implemented the KeyManagement functions Crypto_KeyElementCopy() and Crypto_KeyCopy()
- Implemented the optimized version of the AES CMAC/ECB using precalculated expanded keys, K1 and K2, which are stored in respective key elements

Module version 1.4.0

2017-12-20

- ASCCRYPTO-312 Fixed known issue: Only one crypto primitive for a crypto service is supported
- Added support for multi-instantiation of hardware and software Crypto Drivers

Module version 1.3.0

2017-12-12

- ASCCRYPTO-283 Fixed known issue: Crypto module generates macros for the RamBlockDataAddresses with & as a starting character which leads to compile errors
- ASCCRYPTO-283 Fixed known issue: Crypto does not declare the key data variables to be used in other modules scope
- Improved AES state machine regarding asynchronous/synchronous handling
- Implemented the HMAC algorithm
- Implemented the AEAD with GCM mode
- ASCCRYPTO-313 Fixed known issue: KeyExchange (ECDH) uses wrong KeyElements

Module version 1.2.2

2017-11-27

- ASCCRYPTO-258 Fixed known issue: If CryptoKeyElementSize was equal to the number of provided values in CryptoKeyElementInitValue, the preprocessor derivative resulting from CryptoKeyElementInitValue during code generation had a missing line splice character. This lead to a compilation error.
- ASCCRYPTO-264 Fixed known issue: Crypto module generates duplicate identifiers for NvMRam block and hence the user cannot use the identifier for accessing the NvMRam block
- ASCCRYPTO-255 Fixed known issue: CMAC verification interprets MAC length as bytes instead of BITS
- ASCCRYPTO-293 Fixed known issue: SIGNATUREGENERATE/VERIFY (EdDSA) changes order of bytes in key

Module version 1.2.1

2017-10-20

- ASCCRYPTO-211 The recommended configuration of the Crypto Driver contains all key elements which are specified in SWS_Csm_01022 and not already present in the pre-configuration
- ASCCRYPTO-200 Fixed known issue: Crypto mainfunction needs to be scheduled even if no async job is configured
- ASCCRYPTO-45 Implemented CTR_DRBG based on AES counter mode (AES CTR_DRBG)
- ASCCRYPTO-244 Fixed known issue: CryptoKeyElementInitValue is parsed as comma-separated byte values given in hexadecimal representation (uint8 array). If initialization value contains less hexadecimal values than configured by CryptoKeyElementSize the array is filled with 0x00 in the end.

- ASCCRYPTO-175 Fixed known issue: Crypto primitives could not be called using CRYPTO_OPERATION-MODE_SINGLECALL and asynchronous processing type

Module version 1.2.0

2017-08-30

- ASCCRYPTO-55 Implemented Elliptic Curve Diffie-Hellman (ECDH)
- ASCCRYPTO-55 Fixed known issue: SHA2-512 and Curve25519 (EdDSA) are included
- ASCCRYPTO-152 Fixed known issue: Key elements can be configured as persistent
- Fixed known issue: Key element initial values are fixed
- ASCCRYPTO-151 Fixed known issue: Initialization of Crypto_SymKeyType is fixed
- Fixed known issue: The compiler error caused by a mismatch in parameter Crypto_Generic_Callback was corrected
- ASCCRYPTO-12 Fixed known issue: Variable assignment was corrected

Module version 1.1.0

2017-08-09

- Added NvM support for KeyManagement

Module version 1.0.0

2017-07-28

- Implemented generic part of the Crypto Driver

2.2. New features

- No new features have been added since the last release.

2.3. Elektrobit-specific enhancements

This chapter lists the enhancements provided by the module.

- This module provides no Elektrobit-specific enhancements.

2.4. Deviations

This chapter lists the deviations of the module from the AUTOSAR standard.

- Return value CRYPTO_E_ENTROPY_EXHAUSTED does not exist

  Description:

  - If the corresponding entropy of a random generate function is exhausted, the function cannot return CRYPTO_E_ENTROPY_EXHAUSTED.

  Rationale:

  - This requirement is not applicable. CRYPTO_E_ENTROPY_EXHAUSTED is not defined. CRYPTO_E_ENTROPY_EXHAUSTION will be returned.

  Requirements:

  - SWS_Crypto_00141
- Check of key element Id not applicable

  Description:

  - Crypto_KeyElementGet checks key element and returns its index if present.

  Rationale:

  - This requirement is not applicable. It conflicts with SWS_Crypto_00140. The function Crypto_KeyElementGet returns CRYPTO_E_KEY_NOT_AVAILABLE and shall additionally report the runtime error CRYPTO_E_RE_KEY_NOT_AVAILABLE to the DET, if development errer detection is enabled. Also see Bugzilla entry https://bugzilla.autosar.org/show_bug.cgi?id=81704.

  Requirements:

  - SWS_Crypto_00087
- The requirement is obsolete

  Description:

  - Crypto_KeyElementGet has to check the value pointed by resultLPtr.

  Rationale:

  - This requirement is obsolete in AUTOSAR Specification of Crypto Driver, 4.3.1.

  Requirements:

  - SWS_Crypto_00093
- The requirement is obsolete

  Description:

  - Crypto_KeyElementIdsGet has to check the value pointed by keyElementIdsPtr.

  Rationale:

  - This requirement is obsolete in AUTOSAR Specification of Crypto Driver, 4.3.1.

  Requirements:

  - SWS_Crypto_00164
- Support input lengths of size zero

  Description:

  - Some algorithms (e.g. GCM) can produce valid output, even if the input size is zero.

  Rationale:

  - Implementation is consistent with Bugzilla entry http://www.autosar.org/bugzilla/show_bug.cgi?id=81483

  Requirements:

  - SWS_Crypto_00142
- Only single call processing for RandomGenerate

  Description:

  - For the RandomGenerate service, Crypto_ProcessJob() only supports the operation mode CRYPTO_OPERATIONMODE_SINGLECALL.

  Rationale:

  - Implementation is consistent with R4.4, see also Bugzilla entry http://www.autosar.org/bugzilla/show_bug.cgi?id=78133

  Requirements:

  - SWS_Crypto_91003
- No support for CryptoKeyElementVirtualTargetRef

  Description:

-
  - The Crypto Driver does not support virtual key elements (CryptoKeyElementVirtualTargetRef).

  Rationale:

  - Currently there is no user request regarding virtual key elements. The SWS is currently under discussion (also see http://www.autosar.org/bugzilla/show_bug.cgi?id=80027).

  Requirements:

  - ECUC_Crypto_00028
- It is not possible to cancel every job directly.

  Description:

  - Crypto_CancelJob() should not wait until cancelation of the job is possible.

  Rationale:

  - This requirement is not applicable. Crypto_CancelJob() should not block the application until cancelation is possible.

  Requirements:

  - SWS_Crypto_00143
- The Crypto_CancelJob() API is inconsistent.

  Description:

  - Crypto_CancelJob() expects the input parameter Crypto_JobType* job instead of Crypto_JobInfoType* job.

  Rationale:

  - This requirement is not applicable. Crypto_JobInfoType* job is only available as a const pointer. It is replaced by SWS_Crypto_00122_CORRECTION.

  Requirements:

  - SWS_Crypto_00122
- Queue requirement only applicable for asynchrounous jobs.

  Description:

  - Crypto_ProcessJob() can only check a queue and return CRYPTO_E_QUEUE_FULL if the job is asynchronous.

  Rationale:

    - This requirement is not applicable. CRYPTO_E_QUEUE_FULL should only be returned if an asynchronous job is passed to Crypto_ProcessJob. It is replaced by SWS_Crypto_00032_CORRECTION.

  Requirements:

    - SWS_Crypto_00032

- Synchronous job will not wait if Crypto Driver Object is busy.

  Description:

    - If Crypto_ProcessJob() waits while the crypto driver object is busy before processing a synchronous job, the application might be blocked.

  Rationale:

    - This requirement is not applicable. It conflicts with SWS_Crypto_00034. A synchronous job shall be rejected if the crypto driver object is busy. Also see Bugzilla entry http://www.autosar.org/bugzilla/show_bug.cgi?id=77372.

  Requirements:

    - SWS_Crypto_00120

- Reference SWS_Crypto_00044 does not exist

  Description:

    - The index of the different key elements from the different crypto services has a wrong reference.

  Rationale:

    - This requirement is not applicable. Referenced requirement SWS_Crypto_00044 does not exist in specification.

  Requirements:

    - SWS_Crypto_00037

- Return value CRYPTO_E_KEY_INVALID does not exist

  Description:

    - If a key is in the state "invalid", crypto services which make use of that key, cannot return CRYPTO_E_KEY_INVALID.

  Rationale:

    - This requirement is not applicable. CRYPTO_E_KEY_INVALID is not defined. CRYPTO_E_KEY_NOT_VALID shall be returned. It's replaced by requirement SWS_Crypto_00039_CORRECTION.

  Requirements:

  - SWS_Crypto_00039

- The parameter versionInfo has to be an out parameter

  Description:

  - Service name Crypto_GetVersionInfo: parameter versionInfo has to be an out parameter.

  Rationale:

  - This requirement is not applicable. The parameter versionInfo has to be an out parameter. It's replaced by requirement SWS_Crypto_91001_CORRECTION.

  Requirements:

  - SWS_Crypto_91001

- Check of key element buffer not applicable

  Description:

  - Crypto_KeyElementGet has to check the value pointed by resultLengthPtr.

  Rationale:

  - This requirement is not applicable. It conflicts with SWS_Crypto_00093 as the only way to check if the buffer is sufficient to store the result is to check the value pointed by resultLengthPtr. Also see Bugzilla entry http://www.autosar.org/bugzilla/show_bug.cgi?id=77804.

  Requirements:

  - SWS_Crypto_00147

- The parameter keyElementIdsLengthPtr has to be an inout parameter

  Description:

  - Service name Crypto_KeyElementIdsGet: keyElementIdsLengthPtr has to be an inout parameter.

  Rationale:

  - This requirement is not applicable. The parameter keyElementIdsLengthPtr has to be an inout parameter. It's replaced by requirement SWS_Crypto_00160_CORRECTION.

  Requirements:

  - SWS_Crypto_00160

- Check of buffer keyElementIds not applicable

  Description:

  - Crypto_KeyElementIdsGet has to check the value pointed by resultLengthPtr.

  Rationale:

  - This requirement is not applicable. It conflicts with SWS_Crypto_00093 as the only way to check if the buffer is sufficient to store the result is to check the value pointed by resultLengthPtr. Also see Bugzilla entry http://www.autosar.org/bugzilla/show_bug.cgi?id=77804.

  Requirements:

  - SWS_Crypto_00163

- The parameter seedPtr does not exist

  Description:

  - Crypto_RandomSeed cannot check seedPtr.

  Rationale:

  - This requirement is not applicable. The parameter seedPtr does not exist. It's replaced by requirement SWS_Crypto_00130_CORRECTION.

  Requirements:

  - SWS_Crypto_00130

- The parameter seedLength does not exist

  Description:

  - Crypto_RandomSeed cannot check seedLength.

  Rationale:

  - This requirement is not applicable. The parameter seedLength does not exist. It's replaced by requirement SWS_Crypto_00131_CORRECTION.

  Requirements:

  - SWS_Crypto_00131

- The parameter pubValueLengthPtr does not exist

  Description:

  - Crypto_KeyExchangeCalcPubVal cannot use pubValueLengthPtr.

  Rationale:

  - This requirement is not applicable. Parameter pubValueLengthPtr does not exist, rename to publicValueLengthPtr. It's replaced by requirement SWS_Crypto_00106_CORRECTION.

  Requirements:

  - SWS_Crypto_00106

- The parameter pubValueLengthPtr does not exist

  Description:

  - Crypto_KeyExchangeCalcPubVal cannot check pubValueLengthPtr.

  Rationale:

  - This requirement is not applicable. Parameter pubValueLengthPtr does not exist, rename to publicValueLengthPtr. It's replaced by requirement SWS_Crypto_00107_CORRECTION.

  Requirements:

  - SWS_Crypto_00107

- The parameter partnerPubValueLength does not exist

  Description:

  - Crypto_KeyExchangeCalcSecret cannot check partnerPubValueLength.

  Rationale:

  - This requirement is not applicable. Parameter partnerPubValueLength does not exist, rename to partnerPublicValueLength. It's replaced by requirement SWS_Crypto_00115_CORRECTION.

  Requirements:

  - SWS_Crypto_00115

- The parameter validateCryptoKeyId does not exist

  Description:

  - If the parameter validateCryptoKeyId is out of range and if default error detection for the Crypto Driver is enabled, the function Crypto_CertificateVerify shall report CRYPTO_E_PARAM_HANDLE to the DET and return E_NOT_OK.

  Rationale:

  - This requirement is not applicable. The parameter validateCryptoKeyId does not exist. It's replaced by requirement SWS_Crypto_00174_CORRECTION.

  Requirements:

  - SWS_Crypto_00174

- CryptoDriverObjectId does not start from zero

  Description:

  - CryptoDriverObjectId shall be consecutive, gapless and shall start from zero.

  Rationale:

  - This requirement is not applicable. It's invalidated by note 'The Ids in the configuration containers shall be consecutive, gapless and shall start from zero'. It's replaced by requirement ECUC_Crypto_00009_CORRECTION.

  Requirements:

  - ECUC_Crypto_00009

- CryptoPrimitiveRef has wrong multiplicity

  Description:

  - CryptoPrimitiveRef shall have multiplicity 1..*.

  Rationale:

  - This requirement is not applicable. The multiplicity 1 prevents the configuration of one single driver object which uses all primitives. It's replaced by requirement ECUC_Crypto_00018_CORRECTION. Also see Bugzilla entry http://www.autosar.org/bugzilla/show_bug.cgi?id=77578.

  Requirements:

  - ECUC_Crypto_00018

- CryptoKeyId does not start from zero

  Description:

  - CryptoKeyId shall be consecutive, gapless and shall start from zero.

  Rationale:

  - This requirement is not applicable. It's invalidated by note 'The Ids in the configuration containers shall be consecutive, gapless and shall start from zero'. It's replaced by requirement ECUC_Crypto_00012_CORRECTION.

  Requirements:

  - ECUC_Crypto_00012

- CryptoKeyElementId does not start from zero

  Description:

  - CryptoKeyElementId shall be consecutive, gapless and shall start from zero.

  Rationale:

  - This requirement is not applicable. Identifier shall have a range starting with 0. It's replaced by requirement ECUC_Crypto_00021_CORRECTION.

  Requirements:

  - ECUC_Crypto_00021

- Crypto_KeyElementGet returns E_NOT_OK if the key is invalid

  Description:

  - Crypto_KeyElementGet will return E_NOT_OK instead of E_OK if the key is invalid without calling the lower layer functions.

  Rationale:

  - The key is already in use. See https://bugzilla.autosar.org/show_bug.cgi?id=79359

  Requirements:

  - SWS_Crypto_91006

- Unclear specification about result length configuration

  Description:

  - The HASH and MACGENERATE services shall truncate the result depending on the configured job->jobPrimitiveInfo->primitiveInfo->resultLength.

  Rationale:

  - This requirement is not applicable. It does not explain the relationship between the configured result length and the outputLength API parameter. Further, the Csm has no requirement that specifies which value shall be stored to the job data structure. Also see Bugzilla entry http://www.autosar.org/bugzilla/show_bug.cgi?id=81356.

  Requirements:

  - SWS_Crypto_00065
- Corrected handling of CRYPTO_E_CANCELED

  Description:

  - If the cancelled job is synchronous, Crypto_ProcessJob() shall return CRYPTO_E_CANCELED.

  Rationale:

  - The requirement has been corrected in R4.3.1 as the wrong function has been named. The return value CRYPTO_E_JOB_CANCELED shall be returned by Crypto_ProcessJob() and not by Crypto_CancelJob(). When Crypto_CancelJob() returns CRYPTO_E_JOB_CANCELED, this has a different meaning, namely that the cancellation had to be postponed. Also see Bugzilla entry https://bugzilla.autosar.org/show_bug.cgi?id=77374.

  Requirements:

  - SWS_Crypto_00144
- It is not possible to report CRYPTO_E_INIT_FAILED to the DET

  Description:

  - The initialization of the Crypto Driver can not fail.

  Rationale:

  - This requirement is not applicable. The Crypto_Init function can not report CRYPTO_E_INIT_FAILED to the DET.

  Requirements:

  - SWS_Crypto_00045
- Key generation functionality not supported.

  Description:

  - The key generation functionality is not supported by this CRypto driver.

  Rationale:

  - This functionality has not yet been requested or needed in any senario.

  Requirements:

  - EB_Crypto_00082

- No support for CryptoKeyElementReadAccess CRYPTO_RA_ENCRYPTED and CryptoKeyElementWriteAccess CRYPTO_WA_ENCRYPTED

  Description:

  - The Crypto Driver does not support encrypted keys. The ENCRYPTED CryptoKeyElementReadAccess and CryptoKeyElementWriteAccess have to be disabled.

  Rationale:

  - These requirements are not applicable because the Crypto Driver does not support encrypted keys. Setting the access of a key element to ENCRYPTED would lead to an invalid configuration.

  Requirements:

  - ECUC_Crypto_00024, ECUC_Crypto_00027
- The requirement is only relevant for crypto hardware.

  Description:

  - Key Management Interface: There is no underlying crypto hardware available to be checked.

  Rationale:

  - This requirement is not applicable. There is no underlying crypto hardware for SW Crypto drivers.

  Requirements:

  - SWS_Crypto_00145
- Key management functions cannot return CRYPTO_E_BUSY

  Description:

  - Key management functions will not return CRYPTO_E_BUSY as specified in return values of key management functions.

  Rationale:

  - The key management functions are not linked to any driver object and therefore cannot check their current state.

  Requirements:

  - SWS_Crypto_00148, SWS_Crypto_00155, SWS_Crypto_00160, SWS_Crypto_91007, SWS_Crypto_91009, SWS_Crypto_91010, SWS_Crypto_91011, SWS_Crypto_00171
- Key generation is not supported

  Description:

- New keys cannot be generated via Crypto_KeyGenerate. The function returns E_NOT_OK if all error checks succeeded.

Rationale:

- The hardware does not provide a key generation functionality.

Requirements:

- SWS_Crypto_00165

2.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

- The Crypto Driver only supports Certificate Parse based on self-descriptive card verifiable (CV) and Certificate Verify using RSA-SSA-PSS described in PKCS #1 v2.2: RSA Cryptography Standard 2012.
- Key generation interface: The function of the key generation interface Crypto_KeyGenerate() is not supported.
- Random service: The random service is only functional when used with a Csm module of version 3.0.-2 or later. This is caused by an unclear handling in the Csm and Crypto AUTOSAR specification (see AUTOSAR Bugzilla RfC #78133).
- The functionality of the following configuration element of the CryptoKeyElement is not supported:
  - CryptoKeyElementVirtualTargetRef
- Crypto Driver Object: This delivery is based on one single Crypto Driver Object.
- Rejected AUTOSAR SWS Requirements: Multiple requirements were rejected due to errors. Requirements: SWS_Crypto_00037, SWS_Crypto_00039, SWS_Crypto_91001, SWS_Crypto_00140, SWS_Crypto_00139, SWS_Crypto_00160, SWS_Crypto_00130, SWS_Crypto_00131, SWS_Crypto_00106, SWS_Crypto_00107, SWS_Crypto_00115, SWS_Crypto_00174, ECUC_Crypto_00009, ECUC_Crypto_00018, ECUC_Crypto_00012, SWS_Crypto_00120, SWS_Crypto_00147, SWS_Crypto_00163
- Reporting of undefined DET errors: For DET errors CRYPTO_E_RE_SMALL_BUFFER, CRYPTO_E_RE_ENTROPY_EXHAUSTED, CRYPTO_E_RE_KEY_NOT_AVALIABLE and CRYPTO_E_RE_KEY_EXTRACT_DENIED, the crypto module currently reports the error codes listed in SWS_Crypto_00043 according to the following mapping: CRYPTO_E_RE_SMALL_BUFFER => CRYPTO_E_SMALL_BUFFER; CRYPTO_E_RE_ENTROPY_EXHAUSTED => CRYPTO_E_ENTROPY_EXHAUSTION; CRYPTO_E_RE_KEY_NOT_AVALIABLE => CRYPTO_E_KEY_NOT_AVAILABLE; CRYPTO_E_RE_KEY_EXTRACT_DENIED => CRYPTO_E_KEY_READ_FAIL

- Multi-instantiation: The multi-instantiation of Hardware and Software Crypto Drivers is only functional when used with a CryIf module of version 1.0.3 or later.
- IV for AEAD with GCM: AEAD with GCM supports only initialization vectors of 96 bits length.
- The configured sizes of CsmSymKeyMaxLength, CsmAsymPrivateKeyMaxLength and CsmAsymPublicKeyMaxLength in the Csm module only affect Crypto if used with a Csm module of version 3.0.8 or later.
- Key derivation interface: The key derivation interface is only functional when used with a CryIf module of version 1.0.6 or later.
- RSAES-OAEP: Due to missing enumeration values in CsmDecryptAlgorithmSecondaryFamily and CsmEncryptAlgorithmSecondaryFamily for hash algorithms in the Csm AUTOSAR specification, it is not possible to configure the hash algorithm used for the RSAES-OAEP encryption and decryption services properly (see also AUTOSAR Bugzilla RfC #81809). Therefore, CsmDecryptAlgorithmSecondaryFamily and CsmEncryptAlgorithmSecondaryFamily should be configured to CRYPTO_ALGOFAM_NOT_SET. The Crypto Driver selects the hash algorithm configured in the appropriate RSAES-OAEP encryption and decryption primitives in the Crypto Driver Object. As a consequence, this allows only one RSAES-OAEP encryption and one RSAES-OAEP decryption primitive.
- SHA1 Primitive: the implementation of SHA1 supports only input lengths < 2^32 bytes.

2.6. Open-source software

`Crypto` does not use open-source software.