Document Title	SWS_SecureOnboardCommunication: Complete Change Documentation 4.3.0 - 4.3.1
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	695

Document Status	Final
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	4.3.1

Table of Contents

1 SWS_SecureOnboardCommunication

1.1 Specification Item ECUC_SecOC_00011

Trace References:

none

Content:

Container Name	SecOCRxPduProcessingSecOCRxPduProcessing
Description	Contains the parameters to configure the RxPdus to be verified by the SecOC module.
Configuration Parameters	

Included parameters:

Included Parameters	
Parameter Name	SWS Item ID
SecOCAuthDataFreshnessLen	ECUC_SecOC_00082
SecOCAuthDataFreshnessStartPosition	ECUC_SecOC_00081
SecOCAuthenticationBuildAttempts	ECUC_SecOC_00079
SecOCAuthenticationVerifyAttempts	ECUC_SecOC_00080
SecOCAuthInfoTxLength	ECUC_SecOC_00034
SecOCDataId	ECUC_SecOC_00030
SecOCFreshnessValueId	ECUC_SecOC_00038
SecOCFreshnessValueLength	ECUC_SecOC_00031
SecOCFreshnessValueTxLength	ECUC_SecOC_00032
SecOCReceptionOverflowStrategy	ECUC_SecOC_00076
SecOCReceptionQueueSize	ECUC_SecOC_00077
SecOCUseAuthDataFreshness	ECUC_SecOC_00083
SecOCVerificationStatusPropagationMode	ECUC_SecOC_00046
SecOCRxAuthServiceConfigRef	ECUC_SecOC_00048
SecOCSameBufferPduRef	ECUC_SecOC_00049

Included containers:

Included Containers		
Container Name	Multiplicity	Scope / Dependency
SecOCRxAuthenticPduLayer	1	This container specifies the Pdu that is transmitted by the SecOC module to the PduR after the Mac was verified.

Included Containers		
Container Name	Multiplicity	Scope / Dependency
SecOCRxPduSecuredArea	0..1	This container specifies an area in the Authentic I-Pdu that will be the input to the Authenticator verification algorithm. If this container does not exist in the configuration the complete Authentic I-Pdu will be the input to the Authenticator verification algorithm.
SecOCRxSecuredPduLayer	1	This container specifies the Pdu that is received by the SecOC module from the PduR. For this Pdu the Mac verification is provided.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77090: [SecOC] Configuration of secured area within a Pdu

  Problem description:

  Currently it is not possible to configure a range/area within one Pdu which shall be secured. This possibility shall be added

  Agreed solution:

  SWS_SecOC:
  See attachment https://www.autosar.org/bugzilla/attachment.cgi?id=4500
  ~change range from 1..2^32-1 to 0..2^32-1

  TPS_SystemTemplate:
  add the following optional attributes to SecureCommunicationProps:
  - securedAreaOffset (PositiveInteger) - This attribute defines the start position (offset in byte) of the area within the payload Pdu which will be secured
  - securedAreaLength (PositiveInteger) - This attribute defines the length in bytes of the area within the payload Pdu which will be secured

  Add the following specification item into chapter 6.3.2 SecuredIPdu

  [TPS_SYST_0XXX1] Secured Area in payload Pdu
  The area within the payload Pdu that is secured is specified by the securedAreaOffset and securedAreaLength. In case that these two attributes are not configured the complete
  payload Pdu is secured.

  [constr_xxx1] Existence of securedAreaOffset and securedAreaLength
  If the securedAreaOffset is defined then the securedAreaLength shall be defined as

well and vice versa.
–Last change on issue 77090 comment 29–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.2 Specification Item ECUC_SecOC_00012

Trace References:

none

Content:

Container Name	SecOCTxPduProcessingSecOCTxPduProcessing
Description	Contains the parameters to configure the TxPdus to be secured by the SecOC module.
Configuration Parameters	

Included parameters:

Included Parameters	
Parameter Name	SWS Item ID
SecOCAuthenticationBuildAttempts	ECUC_SecOC_00079
SecOCAuthInfoTxLength	ECUC_SecOC_00018
SecOCDataId	ECUC_SecOC_00014
SecOCFreshnessValueId	ECUC_SecOC_00021
SecOCFreshnessValueLength	ECUC_SecOC_00015
SecOCFreshnessValueTxLength	ECUC_SecOC_00016
SecOCProvideTxTruncatedFreshnessValue	ECUC_SecOC_00084
SecOCUseTxConfirmation	ECUC_SecOC_00085
SecOCSameBufferPduRef	ECUC_SecOC_00010
SecOCTxAuthServiceConfigRef	ECUC_SecOC_00013

Included containers:

Included Containers		
Container Name	Multiplicity	Scope / Dependency
SecOCTxAuthenticPduLayer	1	This container specifies the Pdu that is received by the SecOC module from the PduR. For this Pdu the Mac generation is provided.

Included Containers		
Container Name	Multiplicity	Scope / Dependency
SecOCTxPduSecuredArea	0..1	This container specifies an area in the Authentic I-Pdu that will be the input to the Authenticator generation algorithm. If this container does not exist in the configuration the complete Authentic I-Pdu will be the input to the Authenticator generation algorithm.
SecOCTxSecuredPduLayer	1	This container specifies the Pdu that is transmitted by the SecOC module to the PduR after the Mac was generated.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77090: [SecOC] Configuration of secured area within a Pdu

  Problem description:

  Currently it is not possible to configure a range/area within one Pdu which shall be secured. This possibility shall be added

  Agreed solution:

  SWS_SecOC:
  See attachment https://www.autosar.org/bugzilla/attachment.cgi?id=4500
  ~change range from 1..2^32-1 to 0..2^32-1

  TPS_SystemTemplate:
  add the following optional attributes to SecureCommunicationProps:
  - securedAreaOffset (PositiveInteger) - This attribute defines the start position (offset in byte) of the area within the payload Pdu which will be secured
  - securedAreaLength (PositiveInteger) - This attribute defines the length in bytes of the area within the payload Pdu which will be secured

  Add the following specification item into chapter 6.3.2 SecuredIPdu

  [TPS_SYST_0XXX1] Secured Area in payload Pdu
  The area within the payload Pdu that is secured is specified by the securedAreaOffset and securedAreaLength. In case that these two attributes are not configured the complete
  payload Pdu is secured.

  [constr_xxx1] Existence of securedAreaOffset and securedAreaLength
  If the securedAreaOffset is defined then the securedAreaLength shall be defined as well and vice versa.
  –Last change on issue 77090 comment 29–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.3 Specification Item ECUC_SecOC_00061

Trace References:

none

Content:

Container Name	SecOCRxAuthenticPduSecOCRxAuthenticPdu
Description	This container specifies the Authetic Pdu PDU (that is received by the SecOC module from the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.
Configuration Parameters	

Included parameters:

Included Parameters	
Parameter Name	SWS Item ID
SecOCAuthPduHeaderLength	ECUC_SecOC_00093
SecOCRxAuthenticPduId	ECUC_SecOC_00062
SecOCRxAuthenticPduRef	ECUC_SecOC_00063

Included containers:

No Included Containers

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77336: [SECOC] Dynamic length PDUs (Container) not possible / clear

  Problem description:

  The SecOC SWS does not make any assumptions or restrictions about dynamic length PDUs. But with parameter SecOCAuthDataFreshnessStartPosition it could be impossible to really use dynamic length in SecOC.

  Additionally in SystemTemplate TPS the constraint constr_3139 talks only about some restrictions when dynamic length IPDUs are used. This would imply dynamic length IPDUs should be possible.

Without dynamic length support it is also not possible to support securing complete IDPUM Containers.
One possible solution could be (if dynamic length should not be supported by SecOC) to add a configuration option per IPDUM Container to send always maximum length (padding with 0) to have the static length again.
If dynamic length shall be supported by SecOC a further problem will be the CAN-FD padding.

Agreed solution:

===
AUTOSAR 4.3.1
===

—————–
SRS SecOC
—————–

* Add new section next to 6.1.3.5 and new requirement [SRS_SecOC_xxxx1] Support of padding at lower layer modules and dynamic length Authentic I-PDUs.
* Description: The SecOC module shall be applicable for the use cases with padding at lower layer modules and with dynamic length Authentic I-PDUs.
* Rationale: At receiver side, received Secured I-PDU containing dynamic length Authentic I-PDU may also conatin padding bytes (added by lower layer modules of sender side, to fit to bus-specific L-PDU length constraints, e.g. CAN FD and FlexRay). In such case, receivers cannot identify number of bytes / byte position of the received payload.
* Use Case: dynamic length PDU on CAN FD and FlexRay
* Dependencies: [SRS_SecOC_00012]
* Supporting Material: -

Note: According to CM, RS Main should be referred. But upstream requirements in current (R4.3.0) SRS SecOC are from RS Features, and appropriate requirements are not available in RS Main.
(If we use RS Features, at least [RS_BRF_01568] [RS_BRF_01649] [RS_BRF_01712] [RS_BRF_01716] [RS_BRF_01752] [RS_BRF_02035] [RS_BRF_02036] [RS_BRF_02037] could be mapped to this requirements)

—————–
SWS SecOC
—————–

a1) Adapt Figure 4 in sec. 7.1.1.1

Change from
< (Figure of "Secured I-PDU = Authentic I-PDU | Freshness Value | Authenticator")
< Figure 4: Secured I-PDU contents
to
> (Figure of "Secured I-PDU = Secured I-PDU Header (optional) | Authentic I-PDU | Freshness Value (optional) | Authenticator")
> Figure 4: Secured I-PDU contents
==> to be done as RfC # 77807, not handled in this RfC.

a2) Add new requirements regarding the Secured I-PDU Header and related behavior

Add new requirement [SWS_SecOC_xxxx2] to define the Secured I-PDU Header
> The Secured I-PDU Header shall indicate the length of the Authentic I-PDU in bytes. The length of the Header shall be configurable by the parameter SecOCAuthPduHeaderLength.
> ()
> Note: the SecOC supports combined usage of authentication data in a separate message (secured PDU collection) and Secured I-PDU Header. Also the SecOC covers dynamic length Authentic I-PDU.

Add new requirement [SWS_SecOC_xxxx3] for behavior at transmission (construction) of Secured I-PDUs
> For a Tx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall add the Secured I-PDU Header to the Secured I-PDU with the length of the Authentic I-PDU within the Secured I-PDU, to handle dynamic Authentic I-PDU.
> ()
> Note: Primary purpose of this Header is to indicate the position of Freshness Value and Authenticator in Secured I-PDUs with dynamic length Authentic I-PDU.
> Also some buses which cannot select arbitrary length of L-PDU (e.g. CAN FD and FlexRay) require this Header, because the position of Freshness Value and Authenticator is not always at the end of the Secured I-PDU, as lower layer modules (e.g. CanIf and FrIf) may add bus-specific padding bytes after processing at SecOC (then the L-PDU containing the Secured I-PDU with padding will be: Secured I-PDU = Secured I-PDU Header | Authentic I-PDU | Freshness Value | Authenticator | Bus-specific padding).

Add new requirement [SWS_SecOC_xxxx4] for behavior at reception of Secured I-PDUs
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall assume Secured I-PDU Header shall be available in the Secured I-PDU, to handle dynamic Authentic I-PDU.
> ()

Add new requirement [SWS_SecOC_xxxx5] for behavior at reception of Secured I-PDUs, the Header tells it's longer than the maximum length of the PDU
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0 and the length of Authentic I-PDU in the Header is longer than configured length (in case of dynamic length IPdus (containing a dynamical length signal), this value indicates the maximum data length) of the Authentic I-PDU, the SecOC module shall discard the I-PDU. In such case with SecOC_StartOfReception, BUFREQ_E_NOT_OK shall be returned (see [SWS_COMTYPE_00012]).
> ()
> Note: SecOC_RxIndication has no return value.

Add new requirement [SWS_SecOC_xxxx6] for behavior at reception of Secured I-PDUs, the Header tells it's shorter than received I-PDU length (ignoring the padding at the end)
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall process Secured I-PDU Header, Authentic I-PDU (with the length specified by the Header), Freshness Value and Authenticator of the Rx Secured I-PDU. The rest of bytes in the Secured I-PDU shall be discarded.
> ()

a3) Add a configuration parameter to SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu, SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxAuthenticPdu, SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPdu and SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu to enable/disable Secured I-PDU Header per I-PDU
* SWS Item: ECUC_SecOC_xxxx1
* Name: SecOCAuthPduHeaderLength
* Description:
* This parameter indicates the length (in bytes) of the Secured I-PDU Header in the

Secured I-PDU. The length of zero means there's no header in the PDU.
* Multiplicity: 0..1
* Range: 0..4
* Default: 0

a4) (removed)

a8) Update layout definition (construction) for Secured I-PDUs

Change [SWS_SecOC_00037]
from
< [SWS_SecOC_00037]
< The SecOC module shall construct the Secured I-PDU by adding the Freshness Value and the Authenticator to the Authentic I-PDU.
< (SRS_SecOC_00006)
< Note: The Freshness Counter and the Authenticator included as part of the Secured I-PDU may be truncated per configuration specific to the identifier of the Secured I-PDU. The scheme for the Secured I-PDU looks as follows:
< SecuredPDU = AuthenticIPDU | FreshnessValue [SecOCFreshnessValueTxLength] | Authenticator [SecOCAuthInfoTxLength]
to
> [SWS_SecOC_00037]
> The SecOC module shall construct the Secured I-PDU by adding the Secured I-PDU Header (optional), the Freshness Value (optional) and the Authenticator to the Authentic I-PDU.
> The scheme for the Secured I-PDU (includes the order in which the contents are structured in the Secured I-PDU) shall be compliant with below:
> SecuredPDU = SecuredIPDUHeader (optional) | AuthenticIPDU | FreshnessValue [SecOCFreshnessValueTxLength] (optional) | Authenticator [SecOCAuthInfoTxLength]
> (SRS_SecOC_00006)
> Note: The Freshness Counter and the Authenticator included as part of the Secured I-PDU may be truncated per configuration specific to the identifier of the Secured I-PDU. Also Freshness Value may be a part of Authentic I-PDU (see [SWS_SecOC_00219]).
==> to be done as RfC # 77807, not handled in this RfC.

a9) Add new constraints and notes after [SWS_SecOC_00219]:
> [constr_xxxx1] All signals before SecOCAuthDataFreshnessStartPosition within the Secured I-PDU shall have static length.

> Note: SecOC can use a part of the Authentic I-PDU as freshness when SecOCUseAuthDataFreshness=true, only if the part of the Authentic I-PDU to be used as the freshness is always available at same position in the Authentic I-PDU.

> [constr_xxxx2] Any container I-PDU which contains multiple contained I-PDUs shall be set SecOCUseAuthDataFreshness=false.
> Note: For container PDUs, normally it cannot be ensured which PDU will be put in which position (depends on various timing and trigger conditions). Therefore, container I-PDUs with multiple contained I-PDUs cannot have FV within the Authentic I-PDU.

a10) Adapt Figure 5
==> to be done as RfC # 77807, not handled in this RfC.

a11) Adapt SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxAuthenticPdu and SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu

Change the description of SecOCRxAuthenticPdu (ECUC_SecOC_00061)
from
< This container specifies the Authetic Pdu that is received by the SecOC module from the PduR.
to
> This container specifies the PDU (that is received by the SecOC module from the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.

Change the description of SecOCTxAuthenticPdu ECUC_SecOC_00072
from
< This container specifies the Authetic Pdu that is transmitted by the SecOC module to the PduR after the Mac was generated.
to
> This container specifies the PDU (that is transmitted by the SecOC module to the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.

————————-

TPS System Template (SysT)

————————-

a5) Add new attribute to SecuredIPdu (Table 6.46) which enables SecOCAu-

thPduHeaderLength>0
* Attribute: useSecuredPduHeader
* Type: SecuredPduHeaderEnum
* Mul.: 0..1
* Kind: attr
* Desc: This attribute defines the size of the header which is inserted into the SecuredIPdu. If this attribute is set to anything but noHeader, the SecuredIPdu contains the Secured I-PDU Header to indicate the length of the AuthenticIPdu. The AuthenticIPdu contains the original payload, i.e. the secured data.

SecuredPduHeaderEnum
- noHeader
- securedPduHeader08Bit
- securedPduHeader16Bit
- securedPduHeader32Bit

Desc: Defines the header which will be inserted into the SecuredIPdu.

a6) Change the description of IPduPort.rxSecurityVerification in Table 6.3: IPduPort:
This attribute defines the bypassing of signature authentication or MAC verification in the receiving ECU.
If not defined or set to true the signature authentication or MAC verification shall be performed for the SecuredIPdu.
If set to false the signature authentication or MAC verification shall not be performed for the SecuredIPdu.

Removed [constr_3139].

TPS_SysT_xxxx2: Setting of useSecuredPduHeader attribute
The useSecuredPduHeader shall be set to a value other than noHeader if the length of the payload Pdu is dynamic and is transmitted over a network which may insert padding bytes depending on the length (e.g. CANFD, Flexray).

Add a note below TPS_SysT_xxxx2:

Please note that the dynamic-length Pdu can be an ISignalIPdu that contains a SystemSignal with dynamicLength set to true. In general it is not possible to run diagnostics on fixed-length Pdus. Therefore, there is a probability that at least a subset of DcmIPdus and UserDefinedIPdus can have dynamic length.

a7) Add upstream mapping between useSecuredPduHeader (SysT) and SecOCAuthPduHeaderLength (EcuC) in C.1.4 SecOc Mapping

b4) Add upstream mapping between rxSecurityVerification (SysT) and SecOCSecuredRxPduVerification (EcuC) in C.1.4 SecOc Mapping
Mapping rule: SecOCSecuredRxPduVerification is True if rxSecurityVerification is not defined, otherwise SecOCSecuredRxPduVerification = rxSecurityVerification

——————-

SRS SecOC

——————-

* Add new section next to 6.1.3.5 (or 6.2.1.1) and new requirement [SRS_SecOC_xxxx2] Support of capability to extract Authentic I-PDU without Authentication
* Description: The SecOC module shall be capable to extract Authentic I-PDU from Secured I-PDU, without Authentication.
* Rationale: SecOC can be used as an extractor of Authentic I-PDU from Secured I-PDU, to enable low latency GW behavior when a part of downstream communication clusters doesn't require authentication of PDUs.
* Use Case: Gateway
* Dependencies: [SRS_SecOC_00025]
* Supporting Material: -

Note: According to CM, RS Main should be referred. But upstream requirments in current (R4.3.0) SRS SecOC are from RS Features, and appropriate requirements are not available in RS Main.
(If we use RS Features, at least [RS_BRF_02035] [RS_BRF_02036] [RS_BRF_02037] could be mapped to this requirements)

——————-

SWS SecOC

——————-

b1) Remove [constr_3139] (not [constr_3193] – sorry, constr_3193 is typo in my comment # 10)

b2) Add new requirements regarding skipped authentication behavior at SecOC (just remove FV/MAC from Secured I-PDU)

* Add new section "Extracting Authentic I-PDU without Authentication at SecOC" or "Skipping Authentication for Secured I-PDUs at SecOC"
* Add new requirement [SWS_SecOC_xxxx7] for behavior of SecOC at reception of Secured I-PDUs without Authentication

> For a Rx Secured I-PDU with SecOCSecuredRxPduVerification=false, the SecOC module shall extract the Authentic I-PDU using the length specified by the Secured I-PDU Header without Authentication.
> ()

b3) Add a configuration parameter to SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu and SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection to control authentication behavior at SecOC

* SWS Item: ECUC_SecOC_xxxx3
* Name: SecOCSecuredRxPduVerification
* Description: This parameter defines whether the signature authentication or MAC verification shall be performed on this Secured I-PDU. If set to false, the SecOC module extracts the Authentic I-PDU from the Secured I-PDU without verification.
* Multiplicity: 1
* Type: EcucBooleanParamDef
* Default value: false
* Post-Build Variant Value: true
* Value Configuration Class:
* Pre-compile time: X All Variants
* Scope / Dependency: scope: local
–Last change on issue 77336 comment 69–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.4 Specification Item ECUC_SecOC_00067

Trace References:

none

Content:

Container Name	SecOCRxSecuredPduCollectionSecOCRxSecuredPduCollection
Description	This container specifies two Pdus that are received by the SecOC module from the PduR and a message linking between them. SecOCRxAuthenticPdu contains the original Authentic I-PDU, i.e. the secured data, and the SecOCRxCryptographicPdu contains the Authenticator, i.e. the actual Authentication Information.
Configuration Parameters	

Included parameters:

No Included Parameters	
Parameter Name	SWS Item ID
SecOCSecuredRxPduVerification	ECUC_SecOC_00092

Included containers:

Included Containers		
Container Name	Multiplicity	Scope / Dependency
SecOCRxAuthenticPdu	1	This container specifies the Authetic Pdu PDU (that is received by the SecOC module from the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.
SecOCRxCryptographicPdu	1	This container specifies the Cryptographic Pdu that is received by the SecOC module from the PduR.
SecOCUseMessageLink	0..1	SecOC links an Authentic I-PDU and Cryptographic I-PDU together by repeating a specific part (Message Linker) of the Authentic I-PDU in the Cryptographic I-PDU.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77336: [SECOC] Dynamic length PDUs (Container) not possible / clear

  Problem description:

  The SecOC SWS does not make any assumptions or restrictions about dynamic length PDUs. But with parameter SecOCAuthDataFreshnessStartPosition it could be impossible to really use dynamic length in SecOC.

  Additionally in SystemTemplate TPS the constraint constr_3139 talks only about some restrictions when dynamic length IPDUs are used. This would imply dynamic length IPDUs should be possible.

  Without dynamic length support it is also not possible to support securing complete IDPUM Containers.
  One possible solution could be (if dynamic length should not be supported by SecOC) to add a configuration option per IPDUM Container to send always maximum length (padding with 0) to have the static length again.
  If dynamic length shall be supported by SecOC a further problem will be the CAN-FD padding.

  Agreed solution:

===
AUTOSAR 4.3.1
===

————————-
SRS SecOC
————————-

* Add new section next to 6.1.3.5 and new requirement [SRS_SecOC_xxxx1] Support of padding at lower layer modules and dynamic length Authentic I-PDUs.
* Description: The SecOC module shall be applicable for the use cases with padding at lower layer modules and with dynamic length Authentic I-PDUs.
* Rationale: At receiver side, received Secured I-PDU containing dynamic length Authentic I-PDU may also conatin padding bytes (added by lower layer modules of sender side, to fit to bus-specific L-PDU length constraints, e.g. CAN FD and FlexRay). In such case, receivers cannot identify number of bytes / byte position of the received payload.
* Use Case: dynamic length PDU on CAN FD and FlexRay
* Dependencies: [SRS_SecOC_00012]
* Supporting Material: -

Note: According to CM, RS Main should be referred. But upstream requirements in current (R4.3.0) SRS SecOC are from RS Features, and appropriate requirements are not available in RS Main.
(If we use RS Features, at least [RS_BRF_01568] [RS_BRF_01649] [RS_BRF_01712] [RS_BRF_01716] [RS_BRF_01752] [RS_BRF_02035] [RS_BRF_02036] [RS_BRF_02037] could be mapped to this requirements)

————————-
SWS SecOC
————————-

a1) Adapt Figure 4 in sec. 7.1.1.1

Change from
< (Figure of "Secured I-PDU = Authentic I-PDU | Freshness Value | Authenticator")
< Figure 4: Secured I-PDU contents
to
> (Figure of "Secured I-PDU = Secured I-PDU Header (optional) | Authentic I-PDU | Freshness Value (optional) | Authenticator")
> Figure 4: Secured I-PDU contents

==> to be done as RfC # 77807, not handled in this RfC.

a2) Add new requirements regarding the Secured I-PDU Header and related behavior

Add new requirement [SWS_SecOC_xxxx2] to define the Secured I-PDU Header
> The Secured I-PDU Header shall indicate the length of the Authentic I-PDU in bytes. The length of the Header shall be configurable by the parameter SecOCAuthPduHeaderLength.
> ()
> Note: the SecOC supports combined usage of authentication data in a separate message (secured PDU collection) and Secured I-PDU Header. Also the SecOC covers dynamic length Authentic I-PDU.

Add new requirement [SWS_SecOC_xxxx3] for behavior at transmission (construction) of Secured I-PDUs
> For a Tx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall add the Secured I-PDU Header to the Secured I-PDU with the length of the Authentic I-PDU within the Secured I-PDU, to handle dynamic Authentic I-PDU.
> ()
> Note: Primary purpose of this Header is to indicate the position of Freshness Value and Authenticator in Secured I-PDUs with dynamic length Authentic I-PDU.
> Also some buses which cannot select arbitrary length of L-PDU (e.g. CAN FD and FlexRay) require this Header, because the position of Freshness Value and Authenticator is not always at the end of the Secured I-PDU, as lower layer modules (e.g. CanIf and FrIf) may add bus-specific padding bytes after processing at SecOC (then the L-PDU containing the Secured I-PDU with padding will be: Secured I-PDU = Secured I-PDU Header | Authentic I-PDU | Freshness Value | Authenticator | Bus-specific padding).

Add new requirement [SWS_SecOC_xxxx4] for behavior at reception of Secured I-PDUs
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall assume Secured I-PDU Header shall be available in the Secured I-PDU, to handle dynamic Authentic I-PDU.
> ()

Add new requirement [SWS_SecOC_xxxx5] for behavior at reception of Se-

cured I-PDUs, the Header tells it's longer than the maximum length of the PDU
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0 and the length of Authentic I-PDU in the Header is longer than configured length (in case of dynamic length IPdus (containing a dynamical length signal), this value indicates the maximum data length) of the Authentic I-PDU, the SecOC module shall discard the I-PDU. In such case with SecOC_StartOfReception, BUFREQ_E_NOT_OK shall be returned (see [SWS_COMTYPE_00012]).
> ()
> Note: SecOC_RxIndication has no return value.

Add new requirement [SWS_SecOC_xxxx6] for behavior at reception of Secured I-PDUs, the Header tells it's shorter than received I-PDU length (ignoring the padding at the end)
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall process Secured I-PDU Header, Authentic I-PDU (with the length specified by the Header), Freshness Value and Authenticator of the Rx Secured I-PDU. The rest of bytes in the Secured I-PDU shall be discarded.
> ()

a3) Add a configuration parameter to SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu, SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxAuthenticPdu, SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPdu and SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu to enable/disable Secured I-PDU Header per I-PDU
* SWS Item: ECUC_SecOC_xxxx1
* Name: SecOCAuthPduHeaderLength
* Description:
* This parameter indicates the length (in bytes) of the Secured I-PDU Header in the Secured I-PDU. The length of zero means there's no header in the PDU.
* Multiplicity: 0..1
* Range: 0..4
* Default: 0

a4) (removed)

a8) Update layout definition (construction) for Secured I-PDUs

Change [SWS_SecOC_00037]
from
< [SWS_SecOC_00037]
< The SecOC module shall construct the Secured I-PDU by adding the Freshness Value and the Authenticator to the Authentic I-PDU.
< (SRS_SecOC_00006)
< Note: The Freshness Counter and the Authenticator included as part of the Secured I-PDU may be truncated per configuration specific to the identifier of the Secured I-PDU. The scheme for the Secured I-PDU looks as follows:
< SecuredPDU = AuthenticIPDU | FreshnessValue [SecOCFreshnessValueTxLength] | Authenticator [SecOCAuthInfoTxLength]
to
> [SWS_SecOC_00037]
> The SecOC module shall construct the Secured I-PDU by adding the Secured I-PDU Header (optional), the Freshness Value (optional) and the Authenticator to the Authentic I-PDU.
> The scheme for the Secured I-PDU (includes the order in which the contents are structured in the Secured I-PDU) shall be compliant with below:
> SecuredPDU = SecuredIPDUHeader (optional) | AuthenticIPDU | FreshnessValue [SecOCFreshnessValueTxLength] (optional) | Authenticator [SecOCAuthInfoTxLength]
> (SRS_SecOC_00006)
> Note: The Freshness Counter and the Authenticator included as part of the Secured I-PDU may be truncated per configuration specific to the identifier of the Secured I-PDU. Also Freshness Value may be a part of Authentic I-PDU (see [SWS_SecOC_00219]).
==> to be done as RfC # 77807, not handled in this RfC.

a9) Add new constraints and notes after [SWS_SecOC_00219]:
> [constr_xxxx1] All signals before SecOCAuthDataFreshnessStartPosition within the Secured I-PDU shall have static length.
> Note: SecOC can use a part of the Authentic I-PDU as freshness when SecOCUseAuthDataFreshness=true, only if the part of the Authentic I-PDU to be used as the freshness is always available at same position in the Authentic I-PDU.

> [constr_xxxx2] Any container I-PDU which contains multiple contained I-PDUs shall be set SecOCUseAuthDataFreshness=false.
> Note: For container PDUs, normally it cannot be ensured which PDU will be put in which position (depends on various timing and trigger conditions). Therefore, container I-PDUs with multiple contained I-PDUs cannot have FV within the Authentic I-PDU.

a10) Adapt Figure 5
==> to be done as RfC # 77807, not handled in this RfC.

a11) Adapt SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxAuthenticPdu and SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu

Change the description of SecOCRxAuthenticPdu (ECUC_SecOC_00061)
from
< This container specifies the Authetic Pdu that is received by the SecOC module from the PduR.
to
> This container specifies the PDU (that is received by the SecOC module from the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.

Change the description of SecOCTxAuthenticPdu ECUC_SecOC_00072
from
< This container specifies the Authetic Pdu that is transmitted by the SecOC module to the PduR after the Mac was generated.
to
> This container specifies the PDU (that is transmitted by the SecOC module to the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.

—————-

TPS System Template (SysT)

—————-

a5) Add new attribute to SecuredIPdu (Table 6.46) which enables SecOCAuthPduHeaderLength>0
* Attribute: useSecuredPduHeader
* Type: SecuredPduHeaderEnum
* Mul.: 0..1
* Kind: attr
* Desc: This attribute defines the size of the header which is inserted into the SecuredIPdu. If this attribute is set to anything but noHeader, the SecuredIPdu contains the Secured I-PDU Header to indicate the length of the AuthenticIPdu. The AuthenticIPdu contains the original payload, i.e. the secured data.

SecuredPduHeaderEnum
- noHeader
- securedPduHeader08Bit
- securedPduHeader16Bit
- securedPduHeader32Bit
Desc: Defines the header which will be inserted into the SecuredIPdu.

a6) Change the description of IPduPort.rxSecurityVerification in Table 6.3: IPduPort:
This attribute defines the bypassing of signature authentication or MAC verification in the receiving ECU.
If not defined or set to true the signature authentication or MAC verification shall be performed for the SecuredIPdu.
If set to false the signature authentication or MAC verification shall not be performed for the SecuredIPdu.

Removed [constr_3139].

TPS_SysT_xxxx2: Setting of useSecuredPduHeader attribute
The useSecuredPduHeader shall be set to a value other than noHeader if the length of the payload Pdu is dynamic and is transmitted over a network which may insert padding bytes depending on the length (e.g. CANFD, Flexray).

Add a note below TPS_SysT_xxxx2:

Please note that the dynamic-length Pdu can be an ISignalIPdu that contains a SystemSignal with dynamicLength set to true. In general it is not possible to run diagnostics on fixed-length Pdus. Therefore, there is a probability that at least a subset of DcmIPdus and UserDefinedIPdus can have dynamic length.

a7) Add upstream mapping between useSecuredPduHeader (SysT) and SecOCAuthPduHeaderLength (EcuC) in C.1.4 SecOc Mapping

b4) Add upstream mapping between rxSecurityVerification (SysT) and SecOCSecuredRxPduVerification (EcuC) in C.1.4 SecOc Mapping
Mapping rule: SecOCSecuredRxPduVerification is True if rxSecurityVerification is not defined, otherwise SecOCSecuredRxPduVerification = rxSecurityVerification

——————-

SRS SecOC

——————-

* Add new section next to 6.1.3.5 (or 6.2.1.1) and new requirement

[SRS_SecOC_xxxx2] Support of capability to extract Authentic I-PDU without Authentication
* Description: The SecOC module shall be capable to extract Authentic I-PDU from Secured I-PDU, without Authentication.
* Rationale: SecOC can be used as an extractor of Authentic I-PDU from Secured I-PDU, to enable low latency GW behavior when a part of downstream communication clusters doesn't require authentication of PDUs.
* Use Case: Gateway
* Dependencies: [SRS_SecOC_00025]
* Supporting Material: -

Note: According to CM, RS Main should be referred. But upstream requirments in current (R4.3.0) SRS SecOC are from RS Features, and appropriate requirements are not available in RS Main.
(If we use RS Features, at least [RS_BRF_02035] [RS_BRF_02036] [RS_BRF_02037] could be mapped to this requirements)

——————-
SWS SecOC
——————-

b1) Remove [constr_3139] (not [constr_3193] – sorry, constr_3193 is typo in my comment # 10)

b2) Add new requirements regarding skipped authentication behavior at SecOC (just remove FV/MAC from Secured I-PDU)

* Add new section "Extracting Authentic I-PDU without Authentication at SecOC" or "Skipping Authentication for Secured I-PDUs at SecOC"
* Add new requirement [SWS_SecOC_xxxx7] for behavior of SecOC at reception of Secured I-PDUs without Authentication
> For a Rx Secured I-PDU with SecOCSecuredRxPduVerification=false, the SecOC module shall extract the Authentic I-PDU using the length specified by the Secured I-PDU Header without Authentication.
> ()

b3) Add a configuration parameter to SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu and SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection to control authentication behavior at SecOC

* SWS Item: ECUC_SecOC_xxxx3
* Name: SecOCSecuredRxPduVerification
* Description: This parameter defines whether the signature authentication or MAC verification shall be performed on this Secured I-PDU. If set to false, the SecOC module extracts the Authentic I-PDU from the Secured I-PDU without verification.
* Multiplicity: 1
* Type: EcucBooleanParamDef
* Default value: false
* Post-Build Variant Value: true
* Value Configuration Class:
* Pre-compile time: X All Variants
* Scope / Dependency: scope: local
–Last change on issue 77336 comment 69–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.5 Specification Item ECUC_SecOC_00069

Trace References:

none

Content:

Container Name	SecOCRxSecuredPduSecOCRxSecuredPdu
Description	This container specifies the Pdu that is received by the SecOC module from the PduR. For this Pdu the Mac verification is provided.
Configuration Parameters	

Included parameters:

Included Parameters	
Parameter Name	SWS Item ID
SecOCAuthPduHeaderLength	ECUC_SecOC_00093
SecOCRxSecuredLayerPduId	ECUC_SecOC_00043
SecOCSecuredRxPduVerification	ECUC_SecOC_00092
SecOCRxSecuredLayerPduRef	ECUC_SecOC_00042

Included containers:

No Included Containers

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77336: [SECOC] Dynamic length PDUs (Container) not possible / clear

  Problem description:

  The SecOC SWS does not make any assumptions or restrictions about dynamic length PDUs. But with parameter SecOCAuthDataFreshnessStartPosition it could be impossible to really use dynamic length in SecOC.

  Additionally in SystemTemplate TPS the constraint constr_3139 talks only about some restrictions when dynamic length IPDUs are used. This would imply dynamic length IPDUs should be possible.

  Without dynamic length support it is also not possible to support securing complete IDPUM Containers.
  One possible solution could be (if dynamic length should not be supported by SecOC) to add a configuration option per IPDUM Container to send always maximum length (padding with 0) to have the static length again.
  If dynamic length shall be supported by SecOC a further problem will be the CAN-FD padding.

  Agreed solution:

  ===
  AUTOSAR 4.3.1
  ===

  ———————-
  SRS SecOC
  ———————-

  * Add new section next to 6.1.3.5 and new requirement [SRS_SecOC_xxxx1] Support of padding at lower layer modules and dynamic length Authentic I-PDUs.
  * Description: The SecOC module shall be applicable for the use cases with padding at lower layer modules and with dynamic length Authentic I-PDUs.
  * Rationale: At receiver side, received Secured I-PDU containing dynamic length Authentic I-PDU may also conatin padding bytes (added by lower layer modules of sender side, to fit to bus-specific L-PDU length constraints, e.g. CAN FD and FlexRay). In such case, receivers cannot identify number of bytes / byte position of the received payload.

* Use Case: dynamic length PDU on CAN FD and FlexRay
* Dependencies: [SRS_SecOC_00012]
* Supporting Material: -

Note: According to CM, RS Main should be referred. But upstream requirements in current (R4.3.0) SRS SecOC are from RS Features, and appropriate requirements are not available in RS Main.
(If we use RS Features, at least [RS_BRF_01568] [RS_BRF_01649] [RS_BRF_01712] [RS_BRF_01716] [RS_BRF_01752] [RS_BRF_02035] [RS_BRF_02036] [RS_BRF_02037] could be mapped to this requirements)

——————-

SWS SecOC

——————-

a1) Adapt Figure 4 in sec. 7.1.1.1

Change from
< (Figure of "Secured I-PDU = Authentic I-PDU | Freshness Value | Authenticator")
< Figure 4: Secured I-PDU contents
to
> (Figure of "Secured I-PDU = Secured I-PDU Header (optional) | Authentic I-PDU | Freshness Value (optional) | Authenticator")
> Figure 4: Secured I-PDU contents
==> to be done as RfC # 77807, not handled in this RfC.

a2) Add new requirements regarding the Secured I-PDU Header and related behavior

Add new requirement [SWS_SecOC_xxxx2] to define the Secured I-PDU Header
> The Secured I-PDU Header shall indicate the length of the Authentic I-PDU in bytes. The length of the Header shall be configurable by the parameter SecOCAuthPduHeaderLength.
> ()
> Note: the SecOC supports combined usage of authentication data in a separate message (secured PDU collection) and Secured I-PDU Header. Also the SecOC covers dynamic length Authentic I-PDU.

Add new requirement [SWS_SecOC_xxxx3] for behavior at transmission (construction) of Secured I-PDUs

> For a Tx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall add the Secured I-PDU Header to the Secured I-PDU with the length of the Authentic I-PDU within the Secured I-PDU, to handle dynamic Authentic I-PDU.
> ()
> Note: Primary purpose of this Header is to indicate the position of Freshness Value and Authenticator in Secured I-PDUs with dynamic length Authentic I-PDU.
> Also some buses which cannot select arbitrary length of L-PDU (e.g. CAN FD and FlexRay) require this Header, because the position of Freshness Value and Authenticator is not always at the end of the Secured I-PDU, as lower layer modules (e.g. CanIf and FrIf) may add bus-specific padding bytes after processing at SecOC (then the L-PDU containing the Secured I-PDU with padding will be: Secured I-PDU = Secured I-PDU Header | Authentic I-PDU | Freshness Value | Authenticator | Bus-specific padding).

Add new requirement [SWS_SecOC_xxxx4] for behavior at reception of Secured I-PDUs
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall assume Secured I-PDU Header shall be available in the Secured I-PDU, to handle dynamic Authentic I-PDU.
> ()

Add new requirement [SWS_SecOC_xxxx5] for behavior at reception of Secured I-PDUs, the Header tells it's longer than the maximum length of the PDU
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0 and the length of Authentic I-PDU in the Header is longer than configured length (in case of dynamic length IPdus (containing a dynamical length signal), this value indicates the maximum data length) of the Authentic I-PDU, the SecOC module shall discard the I-PDU. In such case with SecOC_StartOfReception, BUFREQ_E_NOT_OK shall be returned (see [SWS_COMTYPE_00012]).
> ()
> Note: SecOC_RxIndication has no return value.

Add new requirement [SWS_SecOC_xxxx6] for behavior at reception of Secured I-PDUs, the Header tells it's shorter than received I-PDU length (ignoring the padding at the end)
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall process Secured I-PDU Header, Authentic I-PDU (with the length specified by the Header), Freshness Value and Authenticator of the Rx Secured I-PDU. The rest of bytes in the Secured I-PDU shall be discarded.

> ()

a3) Add a configuration parameter to SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu, SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxAuthenticPdu, SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPdu and SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu to enable/disable Secured I-PDU Header per I-PDU
* SWS Item: ECUC_SecOC_xxxx1
* Name: SecOCAuthPduHeaderLength
* Description:
* This parameter indicates the length (in bytes) of the Secured I-PDU Header in the Secured I-PDU. The length of zero means there's no header in the PDU.
* Multiplicity: 0..1
* Range: 0..4
* Default: 0

a4) (removed)

a8) Update layout definition (construction) for Secured I-PDUs

Change [SWS_SecOC_00037]
from
< [SWS_SecOC_00037]
< The SecOC module shall construct the Secured I-PDU by adding the Freshness Value and the Authenticator to the Authentic I-PDU.
< (SRS_SecOC_00006)
< Note: The Freshness Counter and the Authenticator included as part of the Secured I-PDU may be truncated per configuration specific to the identifier of the Secured I-PDU. The scheme for the Secured I-PDU looks as follows:
< SecuredPDU = AuthenticIPDU | FreshnessValue [SecOCFreshnessValueTxLength] | Authenticator [SecOCAuthInfoTxLength]
to
> [SWS_SecOC_00037]
> The SecOC module shall construct the Secured I-PDU by adding the Secured I-PDU Header (optional), the Freshness Value (optional) and the Authenticator to the Authentic I-PDU.
> The scheme for the Secured I-PDU (includes the order in which the contents are

structured in the Secured I-PDU) shall be compliant with below:
> SecuredPDU = SecuredIPDUHeader (optional) | AuthenticIPDU | FreshnessValue [SecOCFreshnessValueTxLength] (optional) | Authenticator [SecOCAuthInfoTxLength]
> (SRS_SecOC_00006)
> Note: The Freshness Counter and the Authenticator included as part of the Secured I-PDU may be truncated per configuration specific to the identifier of the Secured I-PDU. Also Freshness Value may be a part of Authentic I-PDU (see [SWS_SecOC_00219]).
==> to be done as RfC # 77807, not handled in this RfC.

a9) Add new constraints and notes after [SWS_SecOC_00219]:
> [constr_xxxx1] All signals before SecOCAuthDataFreshnessStartPosition within the Secured I-PDU shall have static length.
> Note: SecOC can use a part of the Authentic I-PDU as freshness when SecOCUseAuthDataFreshness=true, only if the part of the Authentic I-PDU to be used as the freshness is always available at same position in the Authentic I-PDU.

> [constr_xxxx2] Any container I-PDU which contains multiple contained I-PDUs shall be set SecOCUseAuthDataFreshness=false.
> Note: For container PDUs, normally it cannot be ensured which PDU will be put in which position (depends on various timing and trigger conditions). Therefore, container I-PDUs with multiple contained I-PDUs cannot have FV within the Authentic I-PDU.

a10) Adapt Figure 5
==> to be done as RfC # 77807, not handled in this RfC.

a11) Adapt SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxAuthenticPdu and SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu

Change the description of SecOCRxAuthenticPdu (ECUC_SecOC_00061)
from
< This container specifies the Authetic Pdu that is received by the SecOC module from the PduR.
to
> This container specifies the PDU (that is received by the SecOC module from the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.

Change the description of SecOCTxAuthenticPdu ECUC_SecOC_00072
from
< This container specifies the Authetic Pdu that is transmitted by the SecOC module to the PduR after the Mac was generated.
to
> This container specifies the PDU (that is transmitted by the SecOC module to the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.

——————-
TPS System Template (SysT)
——————-

a5) Add new attribute to SecuredIPdu (Table 6.46) which enables SecOCAuthPduHeaderLength>0
* Attribute: useSecuredPduHeader
* Type: SecuredPduHeaderEnum
* Mul.: 0..1
* Kind: attr
* Desc: This attribute defines the size of the header which is inserted into the SecuredIPdu. If this attribute is set to anything but noHeader, the SecuredIPdu contains the Secured I-PDU Header to indicate the length of the AuthenticIPdu. The AuthenticIPdu contains the original payload, i.e. the secured data.

SecuredPduHeaderEnum
- noHeader
- securedPduHeader08Bit
- securedPduHeader16Bit
- securedPduHeader32Bit
Desc: Defines the header which will be inserted into the SecuredIPdu.

a6) Change the description of IPduPort.rxSecurityVerification in Table 6.3: IPduPort:
This attribute defines the bypassing of signature authentication or MAC verification in the receiving ECU.
If not defined or set to true the signature authentication or MAC verification shall be performed for the SecuredIPdu.
If set to false the signature authentication or MAC verification shall not be performed for the SecuredIPdu.

Removed [constr_3139].

TPS_SysT_xxxx2: Setting of useSecuredPduHeader attribute

The useSecuredPduHeader shall be set to a value other than noHeader if the length of the payload Pdu is dynamic and is transmitted over a network which may insert padding bytes depending on the length (e.g. CANFD, Flexray).

Add a note below TPS_SysT_xxxx2:

Please note that the dynamic-length Pdu can be an ISignalIPdu that contains a SystemSignal with dynamicLength set to true. In general it is not possible to run diagnostics on fixed-length Pdus. Therefore, there is a probability that at least a subset of DcmIPdus and UserDefinedIPdus can have dynamic length.

a7) Add upstream mapping between useSecuredPduHeader (SysT) and SecOCAuthPduHeaderLength (EcuC) in C.1.4 SecOc Mapping

b4) Add upstream mapping between rxSecurityVerification (SysT) and SecOCSecuredRxPduVerification (EcuC) in C.1.4 SecOc Mapping
Mapping rule: SecOCSecuredRxPduVerification is True if rxSecurityVerification is not defined, otherwise SecOCSecuredRxPduVerification = rxSecurityVerification

——————-
SRS SecOC
——————-

* Add new section next to 6.1.3.5 (or 6.2.1.1) and new requirement [SRS_SecOC_xxxx2] Support of capability to extract Authentic I-PDU without Authentication
* Description: The SecOC module shall be capable to extract Authentic I-PDU from Secured I-PDU, without Authentication.
* Rationale: SecOC can be used as an extractor of Authentic I-PDU from Secured I-PDU, to enable low latency GW behavior when a part of downstream communication clusters doesn't require authentication of PDUs.
* Use Case: Gateway
* Dependencies: [SRS_SecOC_00025]
* Supporting Material: -

Note: According to CM, RS Main should be referred. But upstream requirments in current (R4.3.0) SRS SecOC are from RS Features, and appropriate requirements are not available in RS Main.
(If we use RS Features, at least [RS_BRF_02035] [RS_BRF_02036] [RS_BRF_02037] could be mapped to this requirements)

—————-

SWS SecOC

—————-

b1) Remove [constr_3139] (not [constr_3193] – sorry, constr_3193 is typo in my comment # 10)

b2) Add new requirements regarding skipped authentication behavior at SecOC (just remove FV/MAC from Secured I-PDU)

* Add new section "Extracting Authentic I-PDU without Authentication at SecOC" or "Skipping Authentication for Secured I-PDUs at SecOC"
* Add new requirement [SWS_SecOC_xxxx7] for behavior of SecOC at reception of Secured I-PDUs without Authentication
> For a Rx Secured I-PDU with SecOCSecuredRxPduVerification=false, the SecOC module shall extract the Authentic I-PDU using the length specified by the Secured I-PDU Header without Authentication.
> ()

b3) Add a configuration parameter to SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu and SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection to control authentication behavior at SecOC

* SWS Item: ECUC_SecOC_xxxx3
* Name: SecOCSecuredRxPduVerification
* Description: This parameter defines whether the signature authentication or MAC verification shall be performed on this Secured I-PDU. If set to false, the SecOC module extracts the Authentic I-PDU from the Secured I-PDU without verification.
* Multiplicity: 1
* Type: EcucBooleanParamDef
* Default value: false
* Post-Build Variant Value: true
* Value Configuration Class:
* Pre-compile time: X All Variants
* Scope / Dependency: scope: local
–Last change on issue 77336 comment 69–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.6 Specification Item ECUC_SecOC_00070

Trace References:

none

Content:

Container Name	SecOCTxSecuredPduSecOCTxSecuredPdu
Description	This container specifies one Pdu that is transmitted by the SecOC module to the PduR after the Mac was generated. This Pdu contains the cryptographic information.
Configuration Parameters	

Included parameters:

Included Parameters	
Parameter Name	SWS Item ID
SecOCAuthPduHeaderLength	ECUC_SecOC_00093
SecOCTxSecuredLayerPduId	ECUC_SecOC_00028
SecOCTxSecuredLayerPduRef	ECUC_SecOC_00027

Included containers:

No Included Containers

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77336: [SECOC] Dynamic length PDUs (Container) not possible / clear

  Problem description:

  The SecOC SWS does not make any assumptions or restrictions about dynamic length PDUs. But with parameter SecOCAuthDataFreshnessStartPosition it could be impossible to really use dynamic length in SecOC.

  Additionally in SystemTemplate TPS the constraint constr_3139 talks only about some restrictions when dynamic length IPDUs are used. This would imply dynamic length IPDUs should be possible.

  Without dynamic length support it is also not possible to support securing complete IDPUM Containers.
  One possible solution could be (if dynamic length should not be supported by SecOC) to add a configuration option per IPDUM Container to send always maximum length (padding with 0) to have the static length again.

If dynamic length shall be supported by SecOC a further problem will be the CAN-FD padding.

Agreed solution:

===
AUTOSAR 4.3.1
===

———————-
SRS SecOC
———————-

* Add new section next to 6.1.3.5 and new requirement [SRS_SecOC_xxxx1] Support of padding at lower layer modules and dynamic length Authentic I-PDUs.
* Description: The SecOC module shall be applicable for the use cases with padding at lower layer modules and with dynamic length Authentic I-PDUs.
* Rationale: At receiver side, received Secured I-PDU containing dynamic length Authentic I-PDU may also conatin padding bytes (added by lower layer modules of sender side, to fit to bus-specific L-PDU length constraints, e.g. CAN FD and FlexRay). In such case, receivers cannot identify number of bytes / byte position of the received payload.
* Use Case: dynamic length PDU on CAN FD and FlexRay
* Dependencies: [SRS_SecOC_00012]
* Supporting Material: -

Note: According to CM, RS Main should be referred. But upstream requirements in current (R4.3.0) SRS SecOC are from RS Features, and appropriate requirements are not available in RS Main.
(If we use RS Features, at least [RS_BRF_01568] [RS_BRF_01649] [RS_BRF_01712] [RS_BRF_01716] [RS_BRF_01752] [RS_BRF_02035] [RS_BRF_02036] [RS_BRF_02037] could be mapped to this requirements)

———————-
SWS SecOC
———————-

a1) Adapt Figure 4 in sec. 7.1.1.1

Change from
< (Figure of "Secured I-PDU = Authentic I-PDU | Freshness Value | Authenticator")
< Figure 4: Secured I-PDU contents

to
> (Figure of "Secured I-PDU = Secured I-PDU Header (optional) | Authentic I-PDU | Freshness Value (optional) | Authenticator")
> Figure 4: Secured I-PDU contents
==> to be done as RfC # 77807, not handled in this RfC.

a2) Add new requirements regarding the Secured I-PDU Header and related behavior

Add new requirement [SWS_SecOC_xxxx2] to define the Secured I-PDU Header
> The Secured I-PDU Header shall indicate the length of the Authentic I-PDU in bytes. The length of the Header shall be configurable by the parameter SecOCAuthPduHeaderLength.
> ()
> Note: the SecOC supports combined usage of authentication data in a separate message (secured PDU collection) and Secured I-PDU Header. Also the SecOC covers dynamic length Authentic I-PDU.

Add new requirement [SWS_SecOC_xxxx3] for behavior at transmission (construction) of Secured I-PDUs
> For a Tx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall add the Secured I-PDU Header to the Secured I-PDU with the length of the Authentic I-PDU within the Secured I-PDU, to handle dynamic Authentic I-PDU.
> ()
> Note: Primary purpose of this Header is to indicate the position of Freshness Value and Authenticator in Secured I-PDUs with dynamic length Authentic I-PDU.
> Also some buses which cannot select arbitrary length of L-PDU (e.g. CAN FD and FlexRay) require this Header, because the position of Freshness Value and Authenticator is not always at the end of the Secured I-PDU, as lower layer modules (e.g. CanIf and FrIf) may add bus-specific padding bytes after processing at SecOC (then the L-PDU containing the Secured I-PDU with padding will be: Secured I-PDU = Secured I-PDU Header | Authentic I-PDU | Freshness Value | Authenticator | Bus-specific padding).

Add new requirement [SWS_SecOC_xxxx4] for behavior at reception of Secured I-PDUs
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall assume Secured I-PDU Header shall be available in the Secured I-PDU, to handle dynamic Authentic I-PDU.

> ()

Add new requirement [SWS_SecOC_xxxx5] for behavior at reception of Secured I-PDUs, the Header tells it's longer than the maximum length of the PDU
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0 and the length of Authentic I-PDU in the Header is longer than configured length (in case of dynamic length IPdus (containing a dynamical length signal), this value indicates the maximum data length) of the Authentic I-PDU, the SecOC module shall discard the I-PDU. In such case with SecOC_StartOfReception, BUFREQ_E_NOT_OK shall be returned (see [SWS_COMTYPE_00012]).
> ()
> Note: SecOC_RxIndication has no return value.

Add new requirement [SWS_SecOC_xxxx6] for behavior at reception of Secured I-PDUs, the Header tells it's shorter than received I-PDU length (ignoring the padding at the end)
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall process Secured I-PDU Header, Authentic I-PDU (with the length specified by the Header), Freshness Value and Authenticator of the Rx Secured I-PDU. The rest of bytes in the Secured I-PDU shall be discarded.
> ()

a3) Add a configuration parameter to SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu, SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxAuthenticPdu, SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPdu and SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu to enable/disable Secured I-PDU Header per I-PDU
* SWS Item: ECUC_SecOC_xxxx1
* Name: SecOCAuthPduHeaderLength
* Description:
* This parameter indicates the length (in bytes) of the Secured I-PDU Header in the Secured I-PDU. The length of zero means there's no header in the PDU.
* Multiplicity: 0..1
* Range: 0..4
* Default: 0

a4) (removed)

a8) Update layout definition (construction) for Secured I-PDUs

Change [SWS_SecOC_00037]
from
< [SWS_SecOC_00037]
< The SecOC module shall construct the Secured I-PDU by adding the Freshness Value and the Authenticator to the Authentic I-PDU.
< (SRS_SecOC_00006)
< Note: The Freshness Counter and the Authenticator included as part of the Secured I-PDU may be truncated per configuration specific to the identifier of the Secured I-PDU. The scheme for the Secured I-PDU looks as follows:
< SecuredPDU = AuthenticIPDU | FreshnessValue [SecOCFreshnessValueTxLength] | Authenticator [SecOCAuthInfoTxLength]
to
> [SWS_SecOC_00037]
> The SecOC module shall construct the Secured I-PDU by adding the Secured I-PDU Header (optional), the Freshness Value (optional) and the Authenticator to the Authentic I-PDU.
> The scheme for the Secured I-PDU (includes the order in which the contents are structured in the Secured I-PDU) shall be compliant with below:
> SecuredPDU = SecuredIPDUHeader (optional) | AuthenticIPDU | FreshnessValue [SecOCFreshnessValueTxLength] (optional) | Authenticator [SecOCAuthInfoTxLength]
> (SRS_SecOC_00006)
> Note: The Freshness Counter and the Authenticator included as part of the Secured I-PDU may be truncated per configuration specific to the identifier of the Secured I-PDU. Also Freshness Value may be a part of Authentic I-PDU (see [SWS_SecOC_00219]).
==> to be done as RfC # 77807, not handled in this RfC.

a9) Add new constraints and notes after [SWS_SecOC_00219]:
> [constr_xxxx1] All signals before SecOCAuthDataFreshnessStartPosition within the Secured I-PDU shall have static length.
> Note: SecOC can use a part of the Authentic I-PDU as freshness when SecOCUseAuthDataFreshness=true, only if the part of the Authentic I-PDU to be used as the freshness is always available at same position in the Authentic I-PDU.

> [constr_xxxx2] Any container I-PDU which contains multiple contained I-PDUs shall be set SecOCUseAuthDataFreshness=false.

> Note: For container PDUs, normally it cannot be ensured which PDU will be put in which position (depends on various timing and trigger conditions). Therefore, container I-PDUs with multiple contained I-PDUs cannot have FV within the Authentic I-PDU.

a10) Adapt Figure 5
==> to be done as RfC # 77807, not handled in this RfC.

a11) Adapt SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxAuthenticPdu and SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu

Change the description of SecOCRxAuthenticPdu (ECUC_SecOC_00061)
from
< This container specifies the Authetic Pdu that is received by the SecOC module from the PduR.
to
> This container specifies the PDU (that is received by the SecOC module from the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.

Change the description of SecOCTxAuthenticPdu ECUC_SecOC_00072
from
< This container specifies the Authetic Pdu that is transmitted by the SecOC module to the PduR after the Mac was generated.
to
> This container specifies the PDU (that is transmitted by the SecOC module to the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.

————————-
TPS System Template (SysT)
————————-

a5) Add new attribute to SecuredIPdu (Table 6.46) which enables SecOCAuthPduHeaderLength>0
* Attribute: useSecuredPduHeader
* Type: SecuredPduHeaderEnum
* Mul.: 0..1
* Kind: attr
* Desc: This attribute defines the size of the header which is inserted into the

SecuredIPdu. If this attribute is set to anything but noHeader, the SecuredIPdu contains the Secured I-PDU Header to indicate the length of the AuthenticIPdu. The AuthenticIPdu contains the original payload, i.e. the secured data.

SecuredPduHeaderEnum
- noHeader
- securedPduHeader08Bit
- securedPduHeader16Bit
- securedPduHeader32Bit
Desc: Defines the header which will be inserted into the SecuredIPdu.

a6) Change the description of IPduPort.rxSecurityVerification in Table 6.3: IPduPort:
This attribute defines the bypassing of signature authentication or MAC verification in the receiving ECU.
If not defined or set to true the signature authentication or MAC verification shall be performed for the SecuredIPdu.
If set to false the signature authentication or MAC verification shall not be performed for the SecuredIPdu.

Removed [constr_3139].

TPS_SysT_xxxx2: Setting of useSecuredPduHeader attribute
The useSecuredPduHeader shall be set to a value other than noHeader if the length of the payload Pdu is dynamic and is transmitted over a network which may insert padding bytes depending on the length (e.g. CANFD, Flexray).

Add a note below TPS_SysT_xxxx2:

Please note that the dynamic-length Pdu can be an ISignalIPdu that contains a SystemSignal with dynamicLength set to true. In general it is not possible to run diagnostics on fixed-length Pdus. Therefore, there is a probability that at least a subset of DcmIPdus and UserDefinedIPdus can have dynamic length.

a7) Add upstream mapping between useSecuredPduHeader (SysT) and SecOCAuthPduHeaderLength (EcuC) in C.1.4 SecOc Mapping

b4) Add upstream mapping between rxSecurityVerification (SysT) and SecOCSecuredRxPduVerification (EcuC) in C.1.4 SecOc Mapping
Mapping rule: SecOCSecuredRxPduVerification is True if rxSecurityVerification is not defined, otherwise SecOCSecuredRxPduVerification = rxSecurityVerification

———————-

SRS SecOC
—————-

* Add new section next to 6.1.3.5 (or 6.2.1.1) and new requirement [SRS_SecOC_xxxx2] Support of capability to extract Authentic I-PDU without Authentication
* Description: The SecOC module shall be capable to extract Authentic I-PDU from Secured I-PDU, without Authentication.
* Rationale: SecOC can be used as an extractor of Authentic I-PDU from Secured I-PDU, to enable low latency GW behavior when a part of downstream communication clusters doesn't require authentication of PDUs.
* Use Case: Gateway
* Dependencies: [SRS_SecOC_00025]
* Supporting Material: -

Note: According to CM, RS Main should be referred. But upstream requirments in current (R4.3.0) SRS SecOC are from RS Features, and appropriate requirements are not available in RS Main.
(If we use RS Features, at least [RS_BRF_02035] [RS_BRF_02036] [RS_BRF_02037] could be mapped to this requirements)

—————-
SWS SecOC
—————-

b1) Remove [constr_3139] (not [constr_3193] – sorry, constr_3193 is typo in my comment # 10)

b2) Add new requirements regarding skipped authentication behavior at SecOC (just remove FV/MAC from Secured I-PDU)

* Add new section "Extracting Authentic I-PDU without Authentication at SecOC" or "Skipping Authentication for Secured I-PDUs at SecOC"
* Add new requirement [SWS_SecOC_xxxx7] for behavior of SecOC at reception of Secured I-PDUs without Authentication
> For a Rx Secured I-PDU with SecOCSecuredRxPduVerification=false, the SecOC module shall extract the Authentic I-PDU using the length specified by the Secured I-PDU Header without Authentication.
> ()

b3) Add a configuration parameter to SecOC/SecOCRxPduProcessing/Sec-

OCRxSecuredPduLayer/SecOCRxSecuredPdu and SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection to control authentication behavior at SecOC

* SWS Item: ECUC_SecOC_xxxx3
* Name: SecOCSecuredRxPduVerification
* Description: This parameter defines whether the signature authentication or MAC verification shall be performed on this Secured I-PDU. If set to false, the SecOC module extracts the Authentic I-PDU from the Secured I-PDU without verification.
* Multiplicity: 1
* Type: EcucBooleanParamDef
* Default value: false
* Post-Build Variant Value: true
* Value Configuration Class:
* Pre-compile time: X All Variants
* Scope / Dependency: scope: local
–Last change on issue 77336 comment 69–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.7 Specification Item ECUC_SecOC_00071

Trace References:

none

Content:

Container Name	SecOCTxSecuredPduCollectionSecOCTxSecuredPduCollection
Description	This container specifies the Pdu that is transmitted by the SecOC module to the PduR after the Mac was generated. Two separate Pdus are transmitted to the PduR: Authentic I-PDU and Cryptographic I-PDU.
Configuration Parameters	

Included parameters:

No Included Parameters

Included containers:

Included Containers		
Container Name	Multiplicity	Scope / Dependency
SecOCTxAuthenticPdu	1	This container specifies the Authetic Pdu PDU (that is transmitted by the SecOC module to the PduRafter the Mac was generated) which contains the Secured I-PDU Header and the Authentic I-PDU.
SecOCTxCryptographicPdu	1	This container specifies the Cryptographic Pdu that is transmitted by the SecOC module to the PduR after the Mac was generated.
SecOCUseMessageLink	0..1	SecOC links an Authentic I-PDU and Cryptographic I-PDU together by repeating a specific part (Message Linker) of the Authentic I-PDU in the Cryptographic I-PDU.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77336: [SECOC] Dynamic length PDUs (Container) not possible / clear

  Problem description:

  The SecOC SWS does not make any assumptions or restrictions about dynamic length PDUs. But with parameter SecOCAuthDataFreshnessStartPosition it could be impossible to really use dynamic length in SecOC.

  Additionally in SystemTemplate TPS the constraint constr_3139 talks only about some restrictions when dynamic length IPDUs are used. This would imply dynamic length IPDUs should be possible.

  Without dynamic length support it is also not possible to support securing complete IDPUM Containers.
  One possible solution could be (if dynamic length should not be supported by SecOC) to add a configuration option per IPDUM Container to send always maximum length (padding with 0) to have the static length again.
  If dynamic length shall be supported by SecOC a further problem will be the CAN-FD padding.

  Agreed solution:

  ==
  AUTOSAR 4.3.1
  ==

  SRS SecOC

———————-

* Add new section next to 6.1.3.5 and new requirement [SRS_SecOC_xxxx1] Support of padding at lower layer modules and dynamic length Authentic I-PDUs.
* Description: The SecOC module shall be applicable for the use cases with padding at lower layer modules and with dynamic length Authentic I-PDUs.
* Rationale: At receiver side, received Secured I-PDU containing dynamic length Authentic I-PDU may also conatin padding bytes (added by lower layer modules of sender side, to fit to bus-specific L-PDU length constraints, e.g. CAN FD and FlexRay). In such case, receivers cannot identify number of bytes / byte position of the received payload.
* Use Case: dynamic length PDU on CAN FD and FlexRay
* Dependencies: [SRS_SecOC_00012]
* Supporting Material: -

Note: According to CM, RS Main should be referred. But upstream requirements in current (R4.3.0) SRS SecOC are from RS Features, and appropriate requirements are not available in RS Main.
(If we use RS Features, at least [RS_BRF_01568] [RS_BRF_01649] [RS_BRF_01712] [RS_BRF_01716] [RS_BRF_01752] [RS_BRF_02035] [RS_BRF_02036] [RS_BRF_02037] could be mapped to this requirements)

———————-
SWS SecOC
———————-

a1) Adapt Figure 4 in sec. 7.1.1.1

Change from
< (Figure of "Secured I-PDU = Authentic I-PDU | Freshness Value | Authenticator")
< Figure 4: Secured I-PDU contents
to
> (Figure of "Secured I-PDU = Secured I-PDU Header (optional) | Authentic I-PDU | Freshness Value (optional) | Authenticator")
> Figure 4: Secured I-PDU contents
==> to be done as RfC # 77807, not handled in this RfC.

a2) Add new requirements regarding the Secured I-PDU Header and related behavior

Add new requirement [SWS_SecOC_xxxx2] to define the Secured I-PDU Header

> The Secured I-PDU Header shall indicate the length of the Authentic I-PDU in bytes. The length of the Header shall be configurable by the parameter SecOCAuthPduHeaderLength.
> ()
> Note: the SecOC supports combined usage of authentication data in a separate message (secured PDU collection) and Secured I-PDU Header. Also the SecOC covers dynamic length Authentic I-PDU.

Add new requirement [SWS_SecOC_xxxx3] for behavior at transmission (construction) of Secured I-PDUs
> For a Tx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall add the Secured I-PDU Header to the Secured I-PDU with the length of the Authentic I-PDU within the Secured I-PDU, to handle dynamic Authentic I-PDU.
> ()
> Note: Primary purpose of this Header is to indicate the position of Freshness Value and Authenticator in Secured I-PDUs with dynamic length Authentic I-PDU.
> Also some buses which cannot select arbitrary length of L-PDU (e.g. CAN FD and FlexRay) require this Header, because the position of Freshness Value and Authenticator is not always at the end of the Secured I-PDU, as lower layer modules (e.g. CanIf and FrIf) may add bus-specific padding bytes after processing at SecOC (then the L-PDU containing the Secured I-PDU with padding will be: Secured I-PDU = Secured I-PDU Header | Authentic I-PDU | Freshness Value | Authenticator | Bus-specific padding).

Add new requirement [SWS_SecOC_xxxx4] for behavior at reception of Secured I-PDUs
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall assume Secured I-PDU Header shall be available in the Secured I-PDU, to handle dynamic Authentic I-PDU.
> ()

Add new requirement [SWS_SecOC_xxxx5] for behavior at reception of Secured I-PDUs, the Header tells it's longer than the maximum length of the PDU
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0 and the length of Authentic I-PDU in the Header is longer than configured length (in case of dynamic length IPdus (containing a dynamical length signal), this value indicates the maximum data length) of the Authentic I-PDU, the SecOC module shall discard the I-PDU. In such case with SecOC_StartOfReception, BUFREQ_E_NOT_OK shall be returned (see [SWS_COMTYPE_00012]).

> ()
> Note: SecOC_RxIndication has no return value.

Add new requirement [SWS_SecOC_xxxx6] for behavior at reception of Secured I-PDUs, the Header tells it's shorter than received I-PDU length (ignoring the padding at the end)
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall process Secured I-PDU Header, Authentic I-PDU (with the length specified by the Header), Freshness Value and Authenticator of the Rx Secured I-PDU. The rest of bytes in the Secured I-PDU shall be discarded.
> ()

a3) Add a configuration parameter to SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu, SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxAuthenticPdu, SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPdu and SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu to enable/disable Secured I-PDU Header per I-PDU
* SWS Item: ECUC_SecOC_xxxx1
* Name: SecOCAuthPduHeaderLength
* Description:
* This parameter indicates the length (in bytes) of the Secured I-PDU Header in the Secured I-PDU. The length of zero means there's no header in the PDU.
* Multiplicity: 0..1
* Range: 0..4
* Default: 0

a4) (removed)

a8) Update layout definition (construction) for Secured I-PDUs

Change [SWS_SecOC_00037]
from
< [SWS_SecOC_00037]
< The SecOC module shall construct the Secured I-PDU by adding the Freshness Value and the Authenticator to the Authentic I-PDU.
< (SRS_SecOC_00006)

< Note: The Freshness Counter and the Authenticator included as part of the Secured I-PDU may be truncated per configuration specific to the identifier of the Secured I-PDU. The scheme for the Secured I-PDU looks as follows:
< SecuredPDU = AuthenticIPDU | FreshnessValue [SecOCFreshnessValueTxLength] | Authenticator [SecOCAuthInfoTxLength]
to
> [SWS_SecOC_00037]
> The SecOC module shall construct the Secured I-PDU by adding the Secured I-PDU Header (optional), the Freshness Value (optional) and the Authenticator to the Authentic I-PDU.
> The scheme for the Secured I-PDU (includes the order in which the contents are structured in the Secured I-PDU) shall be compliant with below:
> SecuredPDU = SecuredIPDUHeader (optional) | AuthenticIPDU | FreshnessValue [SecOCFreshnessValueTxLength] (optional) | Authenticator [SecOCAuthInfoTxLength]
> (SRS_SecOC_00006)
> Note: The Freshness Counter and the Authenticator included as part of the Secured I-PDU may be truncated per configuration specific to the identifier of the Secured I-PDU. Also Freshness Value may be a part of Authentic I-PDU (see [SWS_SecOC_00219]).
==> to be done as RfC # 77807, not handled in this RfC.

a9) Add new constraints and notes after [SWS_SecOC_00219]:
> [constr_xxxx1] All signals before SecOCAuthDataFreshnessStartPosition within the Secured I-PDU shall have static length.
> Note: SecOC can use a part of the Authentic I-PDU as freshness when SecOCUseAuthDataFreshness=true, only if the part of the Authentic I-PDU to be used as the freshness is always available at same position in the Authentic I-PDU.

> [constr_xxxx2] Any container I-PDU which contains multiple contained I-PDUs shall be set SecOCUseAuthDataFreshness=false.
> Note: For container PDUs, normally it cannot be ensured which PDU will be put in which position (depends on various timing and trigger conditions). Therefore, container I-PDUs with multiple contained I-PDUs cannot have FV within the Authentic I-PDU.

a10) Adapt Figure 5
==> to be done as RfC # 77807, not handled in this RfC.

a11) Adapt SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxAuthenticPdu and SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu

Change the description of SecOCRxAuthenticPdu (ECUC_SecOC_00061)
from
< This container specifies the Authetic Pdu that is received by the SecOC module from the PduR.
to
> This container specifies the PDU (that is received by the SecOC module from the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.

Change the description of SecOCTxAuthenticPdu ECUC_SecOC_00072
from
< This container specifies the Authetic Pdu that is transmitted by the SecOC module to the PduR after the Mac was generated.
to
> This container specifies the PDU (that is transmitted by the SecOC module to the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.

———————-
TPS System Template (SysT)
———————-

a5) Add new attribute to SecuredIPdu (Table 6.46) which enables SecOCAuthPduHeaderLength>0
* Attribute: useSecuredPduHeader
* Type: SecuredPduHeaderEnum
* Mul.: 0..1
* Kind: attr
* Desc: This attribute defines the size of the header which is inserted into the SecuredIPdu. If this attribute is set to anything but noHeader, the SecuredIPdu contains the Secured I-PDU Header to indicate the length of the AuthenticIPdu. The AuthenticIPdu contains the original payload, i.e. the secured data.

SecuredPduHeaderEnum
- noHeader
- securedPduHeader08Bit
- securedPduHeader16Bit
- securedPduHeader32Bit
Desc: Defines the header which will be inserted into the SecuredIPdu.

a6) Change the description of IPduPort.rxSecurityVerification in Table 6.3: IPduPort:
This attribute defines the bypassing of signature authentication or MAC verification in the receiving ECU.
If not defined or set to true the signature authentication or MAC verification shall be performed for the SecuredIPdu.
If set to false the signature authentication or MAC verification shall not be performed for the SecuredIPdu.

Removed [constr_3139].

TPS_SysT_xxxx2: Setting of useSecuredPduHeader attribute
The useSecuredPduHeader shall be set to a value other than noHeader if the length of the payload Pdu is dynamic and is transmitted over a network which may insert padding bytes depending on the length (e.g. CANFD, Flexray).

Add a note below TPS_SysT_xxxx2:

Please note that the dynamic-length Pdu can be an ISignalIPdu that contains a SystemSignal with dynamicLength set to true. In general it is not possible to run diagnostics on fixed-length Pdus. Therefore, there is a probability that at least a subset of DcmIPdus and UserDefinedIPdus can have dynamic length.

a7) Add upstream mapping between useSecuredPduHeader (SysT) and SecOCAuthPduHeaderLength (EcuC) in C.1.4 SecOc Mapping

b4) Add upstream mapping between rxSecurityVerification (SysT) and SecOCSecuredRxPduVerification (EcuC) in C.1.4 SecOc Mapping
Mapping rule: SecOCSecuredRxPduVerification is True if rxSecurityVerification is not defined, otherwise SecOCSecuredRxPduVerification = rxSecurityVerification

———————-
SRS SecOC
———————-

* Add new section next to 6.1.3.5 (or 6.2.1.1) and new requirement [SRS_SecOC_xxxx2] Support of capability to extract Authentic I-PDU without Authentication
* Description: The SecOC module shall be capable to extract Authentic I-PDU from Secured I-PDU, without Authentication.
* Rationale: SecOC can be used as an extractor of Authentic I-PDU from Secured I-PDU, to enable low latency GW behavior when a part of downstream communication clusters doesn't require authentication of PDUs.

* Use Case: Gateway
* Dependencies: [SRS_SecOC_00025]
* Supporting Material: -

Note: According to CM, RS Main should be referred. But upstream requirments in current (R4.3.0) SRS SecOC are from RS Features, and appropriate requirements are not available in RS Main.
(If we use RS Features, at least [RS_BRF_02035] [RS_BRF_02036] [RS_BRF_02037] could be mapped to this requirements)

—————-

SWS SecOC

—————-

b1) Remove [constr_3139] (not [constr_3193] – sorry, constr_3193 is typo in my comment # 10)

b2) Add new requirements regarding skipped authentication behavior at SecOC (just remove FV/MAC from Secured I-PDU)

* Add new section "Extracting Authentic I-PDU without Authentication at SecOC" or "Skipping Authentication for Secured I-PDUs at SecOC"
* Add new requirement [SWS_SecOC_xxxx7] for behavior of SecOC at reception of Secured I-PDUs without Authentication
> For a Rx Secured I-PDU with SecOCSecuredRxPduVerification=false, the SecOC module shall extract the Authentic I-PDU using the length specified by the Secured I-PDU Header without Authentication.
> ()

b3) Add a configuration parameter to SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu and SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection to control authentication behavior at SecOC

* SWS Item: ECUC_SecOC_xxxx3
* Name: SecOCSecuredRxPduVerification
* Description: This parameter defines whether the signature authentication or MAC verification shall be performed on this Secured I-PDU. If set to false, the SecOC module extracts the Authentic I-PDU from the Secured I-PDU without verification.
* Multiplicity: 1
* Type: EcucBooleanParamDef

* Default value: false
* Post-Build Variant Value: true
* Value Configuration Class:
* Pre-compile time: X All Variants
* Scope / Dependency: scope: local
–Last change on issue 77336 comment 69–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.8 Specification Item ECUC_SecOC_00072

Trace References:

none

Content:

Container Name	SecOCTxAuthenticPduSecOCTxAuthenticPdu
Description	This container specifies the Authetic Pdu PDU (that is transmitted by the SecOC module to the PduRafter the Mac was generated) which contains the Secured I-PDU Header and the Authentic I-PDU.
Configuration Parameters	

Included parameters:

Included Parameters	
Parameter Name	SWS Item ID
SecOCAuthPduHeaderLength	ECUC_SecOC_00093
SecOCTxAuthenticPduId	ECUC_SecOC_00055
SecOCTxAuthenticPduRef	ECUC_SecOC_00056

Included containers:

No Included Containers

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77336: [SECOC] Dynamic length PDUs (Container) not possible / clear

  Problem description:

The SecOC SWS does not make any assumptions or restrictions about dynamic length PDUs. But with parameter SecOCAuthDataFreshnessStartPosition it could be impossible to really use dynamic length in SecOC.

Additionally in SystemTemplate TPS the constraint constr_3139 talks only about some restrictions when dynamic length IPDUs are used. This would imply dynamic length IPDUs should be possible.

Without dynamic length support it is also not possible to support securing complete IDPUM Containers.
One possible solution could be (if dynamic length should not be supported by SecOC) to add a configuration option per IPDUM Container to send always maximum length (padding with 0) to have the static length again.
If dynamic length shall be supported by SecOC a further problem will be the CAN-FD padding.

Agreed solution:

==
AUTOSAR 4.3.1
==

———————-
SRS SecOC
———————-

* Add new section next to 6.1.3.5 and new requirement [SRS_SecOC_xxxx1] Support of padding at lower layer modules and dynamic length Authentic I-PDUs.
* Description: The SecOC module shall be applicable for the use cases with padding at lower layer modules and with dynamic length Authentic I-PDUs.
* Rationale: At receiver side, received Secured I-PDU containing dynamic length Authentic I-PDU may also conatin padding bytes (added by lower layer modules of sender side, to fit to bus-specific L-PDU length constraints, e.g. CAN FD and FlexRay). In such case, receivers cannot identify number of bytes / byte position of the received payload.
* Use Case: dynamic length PDU on CAN FD and FlexRay
* Dependencies: [SRS_SecOC_00012]
* Supporting Material: -

Note: According to CM, RS Main should be referred. But upstream requirements in current (R4.3.0) SRS SecOC are from RS Features, and appropriate requirements are not available in RS Main.
(If we use RS Features, at least [RS_BRF_01568] [RS_BRF_01649]

[RS_BRF_01712] [RS_BRF_01716] [RS_BRF_01752] [RS_BRF_02035] [RS_BRF_02036] [RS_BRF_02037] could be mapped to this requirements)

———————-
SWS SecOC
———————-

a1) Adapt Figure 4 in sec. 7.1.1.1

Change from
< (Figure of "Secured I-PDU = Authentic I-PDU | Freshness Value | Authenticator")
< Figure 4: Secured I-PDU contents
to
> (Figure of "Secured I-PDU = Secured I-PDU Header (optional) | Authentic I-PDU | Freshness Value (optional) | Authenticator")
> Figure 4: Secured I-PDU contents
==> to be done as RfC # 77807, not handled in this RfC.

a2) Add new requirements regarding the Secured I-PDU Header and related behavior

Add new requirement [SWS_SecOC_xxxx2] to define the Secured I-PDU Header
> The Secured I-PDU Header shall indicate the length of the Authentic I-PDU in bytes. The length of the Header shall be configurable by the parameter SecOCAuthPduHeaderLength.
> ()
> Note: the SecOC supports combined usage of authentication data in a separate message (secured PDU collection) and Secured I-PDU Header. Also the SecOC covers dynamic length Authentic I-PDU.

Add new requirement [SWS_SecOC_xxxx3] for behavior at transmission (construction) of Secured I-PDUs
> For a Tx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall add the Secured I-PDU Header to the Secured I-PDU with the length of the Authentic I-PDU within the Secured I-PDU, to handle dynamic Authentic I-PDU.
> ()
> Note: Primary purpose of this Header is to indicate the position of Freshness Value and Authenticator in Secured I-PDUs with dynamic length Authentic I-PDU.
> Also some buses which cannot select arbitrary length of L-PDU (e.g. CAN FD

and FlexRay) require this Header, because the position of Freshness Value and Authenticator is not always at the end of the Secured I-PDU, as lower layer modules (e.g. CanIf and FrIf) may add bus-specific padding bytes after processing at SecOC (then the L-PDU containing the Secured I-PDU with padding will be: Secured I-PDU = Secured I-PDU Header | Authentic I-PDU | Freshness Value | Authenticator | Bus-specific padding).

Add new requirement [SWS_SecOC_xxxx4] for behavior at reception of Secured I-PDUs
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall assume Secured I-PDU Header shall be available in the Secured I-PDU, to handle dynamic Authentic I-PDU.
> ()

Add new requirement [SWS_SecOC_xxxx5] for behavior at reception of Secured I-PDUs, the Header tells it's longer than the maximum length of the PDU
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0 and the length of Authentic I-PDU in the Header is longer than configured length (in case of dynamic length IPdus (containing a dynamical length signal), this value indicates the maximum data length) of the Authentic I-PDU, the SecOC module shall discard the I-PDU. In such case with SecOC_StartOfReception, BUFREQ_E_NOT_OK shall be returned (see [SWS_COMTYPE_00012]).
> ()
> Note: SecOC_RxIndication has no return value.

Add new requirement [SWS_SecOC_xxxx6] for behavior at reception of Secured I-PDUs, the Header tells it's shorter than received I-PDU length (ignoring the padding at the end)
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall process Secured I-PDU Header, Authentic I-PDU (with the length specified by the Header), Freshness Value and Authenticator of the Rx Secured I-PDU. The rest of bytes in the Secured I-PDU shall be discarded.
> ()

a3) Add a configuration parameter to SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu, SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxAuthenticPdu, SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPdu and Sec-

OC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu to enable/disable Secured I-PDU Header per I-PDU
* SWS Item: ECUC_SecOC_xxxx1
* Name: SecOCAuthPduHeaderLength
* Description:
* This parameter indicates the length (in bytes) of the Secured I-PDU Header in the Secured I-PDU. The length of zero means there's no header in the PDU.
* Multiplicity: 0..1
* Range: 0..4
* Default: 0

a4) (removed)

a8) Update layout definition (construction) for Secured I-PDUs

Change [SWS_SecOC_00037]
from
< [SWS_SecOC_00037]
< The SecOC module shall construct the Secured I-PDU by adding the Freshness Value and the Authenticator to the Authentic I-PDU.
< (SRS_SecOC_00006)
< Note: The Freshness Counter and the Authenticator included as part of the Secured I-PDU may be truncated per configuration specific to the identifier of the Secured I-PDU. The scheme for the Secured I-PDU looks as follows:
< SecuredPDU = AuthenticIPDU | FreshnessValue [SecOCFreshnessValueTxLength] | Authenticator [SecOCAuthInfoTxLength]
to
> [SWS_SecOC_00037]
> The SecOC module shall construct the Secured I-PDU by adding the Secured I-PDU Header (optional), the Freshness Value (optional) and the Authenticator to the Authentic I-PDU.
> The scheme for the Secured I-PDU (includes the order in which the contents are structured in the Secured I-PDU) shall be compliant with below:
> SecuredPDU = SecuredIPDUHeader (optional) | AuthenticIPDU | FreshnessValue [SecOCFreshnessValueTxLength] (optional) | Authenticator [SecOCAuthInfoTxLength]
> (SRS_SecOC_00006)
> Note: The Freshness Counter and the Authenticator included as part of the Secured I-PDU may be truncated per configuration specific to the identifier of the Secured I-PDU. Also Freshness Value may be a part of Authentic I-PDU (see

[SWS_SecOC_00219]).
==> to be done as RfC # 77807, not handled in this RfC.

a9) Add new constraints and notes after [SWS_SecOC_00219]:
> [constr_xxxx1] All signals before SecOCAuthDataFreshnessStartPosition within the Secured I-PDU shall have static length.
> Note: SecOC can use a part of the Authentic I-PDU as freshness when SecOCUseAuthDataFreshness=true, only if the part of the Authentic I-PDU to be used as the freshness is always available at same position in the Authentic I-PDU.

> [constr_xxxx2] Any container I-PDU which contains multiple contained I-PDUs shall be set SecOCUseAuthDataFreshness=false.
> Note: For container PDUs, normally it cannot be ensured which PDU will be put in which position (depends on various timing and trigger conditions). Therefore, container I-PDUs with multiple contained I-PDUs cannot have FV within the Authentic I-PDU.

a10) Adapt Figure 5
==> to be done as RfC # 77807, not handled in this RfC.

a11) Adapt SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxAuthenticPdu and SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu

Change the description of SecOCRxAuthenticPdu (ECUC_SecOC_00061)
from
< This container specifies the Authetic Pdu that is received by the SecOC module from the PduR.
to
> This container specifies the PDU (that is received by the SecOC module from the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.

Change the description of SecOCTxAuthenticPdu ECUC_SecOC_00072
from
< This container specifies the Authetic Pdu that is transmitted by the SecOC module to the PduR after the Mac was generated.
to
> This container specifies the PDU (that is transmitted by the SecOC module to the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.

——————-

TPS System Template (SysT)

——————-

a5) Add new attribute to SecuredIPdu (Table 6.46) which enables SecOCAuthPduHeaderLength>0
* Attribute: useSecuredPduHeader
* Type: SecuredPduHeaderEnum
* Mul.: 0..1
* Kind: attr
* Desc: This attribute defines the size of the header which is inserted into the SecuredIPdu. If this attribute is set to anything but noHeader, the SecuredIPdu contains the Secured I-PDU Header to indicate the length of the AuthenticIPdu. The AuthenticIPdu contains the original payload, i.e. the secured data.

SecuredPduHeaderEnum
- noHeader
- securedPduHeader08Bit
- securedPduHeader16Bit
- securedPduHeader32Bit
Desc: Defines the header which will be inserted into the SecuredIPdu.

a6) Change the description of IPduPort.rxSecurityVerification in Table 6.3: IPduPort:
This attribute defines the bypassing of signature authentication or MAC verification in the receiving ECU.
If not defined or set to true the signature authentication or MAC verification shall be performed for the SecuredIPdu.
If set to false the signature authentication or MAC verification shall not be performed for the SecuredIPdu.

Removed [constr_3139].

TPS_SysT_xxxx2: Setting of useSecuredPduHeader attribute
The useSecuredPduHeader shall be set to a value other than noHeader if the length of the payload Pdu is dynamic and is transmitted over a network which may insert padding bytes depending on the length (e.g. CANFD, Flexray).

Add a note below TPS_SysT_xxxx2:

Please note that the dynamic-length Pdu can be an ISignalIPdu that contains a SystemSignal with dynamicLength set to true. In general it is not possible to run diagnostics on fixed-length Pdus. Therefore, there is a probability that at least a subset of DcmIPdus and UserDefinedIPdus can have dynamic length.

a7) Add upstream mapping between useSecuredPduHeader (SysT) and SecOCAuthPduHeaderLength (EcuC) in C.1.4 SecOc Mapping

b4) Add upstream mapping between rxSecurityVerification (SysT) and SecOCSecuredRxPduVerification (EcuC) in C.1.4 SecOc Mapping
Mapping rule: SecOCSecuredRxPduVerification is True if rxSecurityVerification is not defined, otherwise SecOCSecuredRxPduVerification = rxSecurityVerification

————————-

SRS SecOC

————————-

* Add new section next to 6.1.3.5 (or 6.2.1.1) and new requirement [SRS_SecOC_xxxx2] Support of capability to extract Authentic I-PDU without Authentication
* Description: The SecOC module shall be capable to extract Authentic I-PDU from Secured I-PDU, without Authentication.
* Rationale: SecOC can be used as an extractor of Authentic I-PDU from Secured I-PDU, to enable low latency GW behavior when a part of downstream communication clusters doesn't require authentication of PDUs.
* Use Case: Gateway
* Dependencies: [SRS_SecOC_00025]
* Supporting Material: -

Note: According to CM, RS Main should be referred. But upstream requirments in current (R4.3.0) SRS SecOC are from RS Features, and appropriate requirements are not available in RS Main.
(If we use RS Features, at least [RS_BRF_02035] [RS_BRF_02036] [RS_BRF_02037] could be mapped to this requirements)

————————-

SWS SecOC

————————-

b1) Remove [constr_3139] (not [constr_3193] – sorry, constr_3193 is typo in my comment # 10) b2) Add new requirements regarding skipped authentication behavior at SecOC (just remove FV/MAC from Secured I-PDU)

* Add new section "Extracting Authentic I-PDU without Authentication at SecOC" or "Skipping Authentication for Secured I-PDUs at SecOC"
* Add new requirement [SWS_SecOC_xxxx7] for behavior of SecOC at reception of Secured I-PDUs without Authentication
> For a Rx Secured I-PDU with SecOCSecuredRxPduVerification=false, the SecOC module shall extract the Authentic I-PDU using the length specified by the Secured I-PDU Header without Authentication.
> ()

b3) Add a configuration parameter to SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu and SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection to control authentication behavior at SecOC

* SWS Item: ECUC_SecOC_xxxx3
* Name: SecOCSecuredRxPduVerification
* Description: This parameter defines whether the signature authentication or MAC verification shall be performed on this Secured I-PDU. If set to false, the SecOC module extracts the Authentic I-PDU from the Secured I-PDU without verification.
* Multiplicity: 1
* Type: EcucBooleanParamDef
* Default value: false
* Post-Build Variant Value: true
* Value Configuration Class:
* Pre-compile time: X All Variants
* Scope / Dependency: scope: local
–Last change on issue 77336 comment 69–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.9 Specification Item ECUC_SecOC_00081

Trace References:

none

Content:

Name	SecOCAuthDataFreshnessStartPositionSecOCRxPduProcessing.SecOCAuthDataFreshness StartPosition		
Parent Container	SecOCRxPduProcessing		
Description	This value determines the start position in bits (uint16) of the Authentic PDU that shall be passed on to the Freshness SWC. The bit position starts counting from the MSB of the first byte of the PDU.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 18446744073709551615 65535		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77297: Which is the correct range of "SecOCAuthDataFreshnessStartPosition"?

  Problem description:

  In the specification of SecOC, ECUC_SecOC_00081 is mentioned as follows:

  This value("SecOCAuthDataFreshnessStartPosition") determines the start position in bits (uint16) of the Authentic PDU that shall be passed on to the Freshness SWC.

  However, as can be seen at ECUC_SecOC_00081, the range of "SecOCAuthDataFreshnessStartPosition" is presented the range of 64bits(0 .. 18446744073709551615). Which is the correct range of "SecOCAuthDataFreshnessStartPosition"?

  Could you please check and correct it?

  Agreed solution:

  At "Range" in [ECUC_SecOC_00081] and [ECUC_SecOC_00082]:

  Replace "0 .. 18446744073709551615" with "0 .. 65535"
  –Last change on issue 77297 comment 11–

  BW-C-Level:

Application	Specification	Bus
1	4	1

1.10 Specification Item ECUC_SecOC_00082

Trace References:

none

Content:

Name	SecOCAuthDataFreshnessLenSecOCRxPduProcessing.SecOCAuthDataFreshnessLen		
Parent Container	SecOCRxPduProcessing		
Description	The length of the external authentic PDU data in bits (uint16).		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 18446744073709551615 65535		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77297: Which is the correct range of "SecOCAuthDataFreshnessStartPosition"?

  Problem description:

  In the specification of SecOC, ECUC_SecOC_00081 is mentioned as follows:

  This value("SecOCAuthDataFreshnessStartPosition") determines the start position in bits (uint16) of the Authentic PDU that shall be passed on to the Freshness SWC.

  However, as can be seen at ECUC_SecOC_00081, the range of "SecOCAuthDataFreshnessStartPosition" is presented the range of 64bits(0 .. 18446744073709551615). Which is the correct range of "SecOCAuthDataFreshnessStartPosition"?

  Could you please check and correct it?

Agreed solution:

At "Range" in [ECUC_SecOC_00081] and [ECUC_SecOC_00082]:

Replace "0 .. 18446744073709551615" with "0 .. 65535"
–Last change on issue 77297 comment 11–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.11 Specification Item ECUC_SecOC_00086

Trace References:

none

Content:

Container Name	SecOCTxPduSecuredAreaSecOCTxPduSecuredArea
Description	This container specifies an area in the Authentic I-Pdu that will be the input to the Authenticator generation algorithm. If this container does not exist in the configuration the complete Authentic I-Pdu will be the input to the Authenticator generation algorithm.
Configuration Parameters	

Included parameters:

Included Parameters	
Parameter Name	SWS Item ID
SecOCSecuredTxPduLength	ECUC_SecOC_00088
SecOCSecuredTxPduOffset	ECUC_SecOC_00087

Included containers:

No Included Containers

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77090: [SecOC] Configuration of secured area within a Pdu

  Problem description:

Currently it is not possible to configure a range/area within one Pdu which shall be secured. This possibility shall be added

Agreed solution:

SWS_SecOC:
See attachment https://www.autosar.org/bugzilla/attachment.cgi?id=4500
~change range from 1..2^32-1 to 0..2^32-1

TPS_SystemTemplate:
add the following optional attributes to SecureCommunicationProps:
- securedAreaOffset (PositiveInteger) - This attribute defines the start position (offset in byte) of the area within the payload Pdu which will be secured
- securedAreaLength (PositiveInteger) - This attribute defines the length in bytes of the area within the payload Pdu which will be secured

Add the following specification item into chapter 6.3.2 SecuredIPdu

[TPS_SYST_0XXX1] Secured Area in payload Pdu
The area within the payload Pdu that is secured is specified by the securedAreaOffset and securedAreaLength. In case that these two attributes are not configured the complete
payload Pdu is secured.

[constr_xxx1] Existence of securedAreaOffset and securedAreaLength
If the securedAreaOffset is defined then the securedAreaLength shall be defined as well and vice versa.
–Last change on issue 77090 comment 29–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.12 Specification Item ECUC_SecOC_00087

Trace References:

none

Content:

Name	SecOCSecuredTxPduOffsetSecOCTxPduSecuredArea.SecOCSecuredTxPduOffsetin container SecOCTxPduSecuredArea

Description	This parameter defines the start position (offset in bytes) of the area within the Pdu which shall be secured		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default value	0		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77090: [SecOC] Configuration of secured area within a Pdu

  Problem description:

  Currently it is not possible to configure a range/area within one Pdu which shall be secured. This possibility shall be added

  Agreed solution:

  SWS_SecOC:
  See attachment https://www.autosar.org/bugzilla/attachment.cgi?id=4500
  ~change range from 1..2^32-1 to 0..2^32-1

  TPS_SystemTemplate:
  add the following optional attributes to SecureCommunicationProps:
  - securedAreaOffset (PositiveInteger) - This attribute defines the start position (offset in byte) of the area within the payload Pdu which will be secured
  - securedAreaLength (PositiveInteger) - This attribute defines the length in bytes of the area within the payload Pdu which will be secured

  Add the following specification item into chapter 6.3.2 SecuredIPdu

  [TPS_SYST_0XXX1] Secured Area in payload Pdu
  The area within the payload Pdu that is secured is specified by the securedAreaOffset and securedAreaLength. In case that these two attributes are not configured the complete
  payload Pdu is secured.

[constr_xxx1] Existence of securedAreaOffset and securedAreaLength
If the securedAreaOffset is defined then the securedAreaLength shall be defined as well and vice versa.
–Last change on issue 77090 comment 29–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.13 Specification Item ECUC_SecOC_00088

Trace References:

none

Content:

Name	SecOCSecuredTxPduLengthSecOCTxPduSecuredArea.SecOCSecuredTxPduLengthin container SecOCTxPduSecuredArea		
Description	This parameter defines the length (in bytes) of the area within the Pdu which shall be secured		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default value	–		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77090: [SecOC] Configuration of secured area within a Pdu

  Problem description:

  Currently it is not possible to configure a range/area within one Pdu which shall be secured. This possibility shall be added

  Agreed solution:

SWS_SecOC:
See attachment https://www.autosar.org/bugzilla/attachment.cgi?id=4500
~change range from 1..2^32-1 to 0..2^32-1

TPS_SystemTemplate:
add the following optional attributes to SecureCommunicationProps:
- securedAreaOffset (PositiveInteger) - This attribute defines the start position (offset in byte) of the area within the payload Pdu which will be secured
- securedAreaLength (PositiveInteger) - This attribute defines the length in bytes of the area within the payload Pdu which will be secured

Add the following specification item into chapter 6.3.2 SecuredIPdu

[TPS_SYST_0XXX1] Secured Area in payload Pdu
The area within the payload Pdu that is secured is specified by the securedAreaOffset and securedAreaLength. In case that these two attributes are not configured the complete
payload Pdu is secured.

[constr_xxx1] Existence of securedAreaOffset and securedAreaLength
If the securedAreaOffset is defined then the securedAreaLength shall be defined as well and vice versa.
–Last change on issue 77090 comment 29–

BW-C-Level:

Application	**Specification**	**Bus**
1	1	1

1.14 Specification Item ECUC_SecOC_00089

Trace References:

none

Content:

Container Name	SecOCRxPduSecuredAreaSecOCRxPduSecuredArea
Description	This container specifies an area in the Authentic I-Pdu that will be the input to the Authenticator verification algorithm. If this container does not exist in the configuration the complete Authentic I-Pdu will be the input to the Authenticator verification algorithm.
Configuration Parameters	

Included parameters:

Included Parameters	
Parameter Name	SWS Item ID
SecOCSecuredRxPduLength	ECUC_SecOC_00091
SecOCSecuredRxPduOffset	ECUC_SecOC_00090

Included containers:

No Included Containers

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77090: [SecOC] Configuration of secured area within a Pdu

  Problem description:

  Currently it is not possible to configure a range/area within one Pdu which shall be secured. This possibility shall be added

  Agreed solution:

  SWS_SecOC:
  See attachment https://www.autosar.org/bugzilla/attachment.cgi?id=4500
  ~change range from 1..2^32-1 to 0..2^32-1

  TPS_SystemTemplate:
  add the following optional attributes to SecureCommunicationProps:
  - securedAreaOffset (PositiveInteger) - This attribute defines the start position (offset in byte) of the area within the payload Pdu which will be secured
  - securedAreaLength (PositiveInteger) - This attribute defines the length in bytes of the area within the payload Pdu which will be secured

  Add the following specification item into chapter 6.3.2 SecuredIPdu

  [TPS_SYST_0XXX1] Secured Area in payload Pdu
  The area within the payload Pdu that is secured is specified by the securedAreaOffset and securedAreaLength. In case that these two attributes are not configured the complete
  payload Pdu is secured.

  [constr_xxx1] Existence of securedAreaOffset and securedAreaLength
  If the securedAreaOffset is defined then the securedAreaLength shall be defined as

well and vice versa.
–Last change on issue 77090 comment 29–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.15 Specification Item ECUC_SecOC_00090

Trace References:

none

Content:

Name	SecOCSecuredRxPduOffsetSecOCRxPduSecuredArea.SecOCSecuredRxPduOffsetin container SecOCRxPduSecuredArea		
Description	This parameter defines the start position (offset in bytes) of the area within the Pdu which is secured		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default value	0		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77090: [SecOC] Configuration of secured area within a Pdu

  Problem description:

  Currently it is not possible to configure a range/area within one Pdu which shall be secured. This possibility shall be added

  Agreed solution:

  SWS_SecOC:
  See attachment https://www.autosar.org/bugzilla/attachment.cgi?id=4500

~change range from 1..2^32-1 to 0..2^32-1

TPS_SystemTemplate:
add the following optional attributes to SecureCommunicationProps:
- securedAreaOffset (PositiveInteger) - This attribute defines the start position (offset in byte) of the area within the payload Pdu which will be secured
- securedAreaLength (PositiveInteger) - This attribute defines the length in bytes of the area within the payload Pdu which will be secured

Add the following specification item into chapter 6.3.2 SecuredIPdu

[TPS_SYST_0XXX1] Secured Area in payload Pdu
The area within the payload Pdu that is secured is specified by the securedAreaOffset and securedAreaLength. In case that these two attributes are not configured the complete
payload Pdu is secured.

[constr_xxx1] Existence of securedAreaOffset and securedAreaLength
If the securedAreaOffset is defined then the securedAreaLength shall be defined as well and vice versa.
–Last change on issue 77090 comment 29–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.16 Specification Item ECUC_SecOC_00091

Trace References:

none

Content:

Name	SecOCSecuredRxPduLengthSecOCRxPduSecuredArea.SecOCSecuredRxPduLengthin container SecOCRxPduSecuredArea	
Description	This parameter defines the length (in bytes) of the area within the Pdu which is secured	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 4294967295	
Default value	–	

Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77090: [SecOC] Configuration of secured area within a Pdu

  Problem description:

  Currently it is not possible to configure a range/area within one Pdu which shall be secured. This possibility shall be added

  Agreed solution:

  SWS_SecOC:
  See attachment https://www.autosar.org/bugzilla/attachment.cgi?id=4500
  ~change range from 1..2^32-1 to 0..2^32-1

  TPS_SystemTemplate:
  add the following optional attributes to SecureCommunicationProps:
  - securedAreaOffset (PositiveInteger) - This attribute defines the start position (offset in byte) of the area within the payload Pdu which will be secured
  - securedAreaLength (PositiveInteger) - This attribute defines the length in bytes of the area within the payload Pdu which will be secured

  Add the following specification item into chapter 6.3.2 SecuredIPdu

  [TPS_SYST_0XXX1] Secured Area in payload Pdu
  The area within the payload Pdu that is secured is specified by the securedAreaOffset and securedAreaLength. In case that these two attributes are not configured the complete
  payload Pdu is secured.

  [constr_xxx1] Existence of securedAreaOffset and securedAreaLength
  If the securedAreaOffset is defined then the securedAreaLength shall be defined as well and vice versa.
  –Last change on issue 77090 comment 29–

  BW-C-Level:

Application	Specification	Bus
1	1	1

1.17 Specification Item ECUC_SecOC_00092

Trace References:

none

Content:

Name	SecOCSecuredRxPduVerificationSecOCRxSecuredPduCollection.SecOCSecuredRxPdu Verification		
Parent Container	SecOCRxSecuredPduCollection		
Description	This parameter defines whether the signature authentication or MAC verification shall be performed on this Secured I-PDU. If set to false, the SecOC module extracts the Authentic I-PDU from the Secured I-PDU without verification.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77336: [SECOC] Dynamic length PDUs (Container) not possible / clear

  Problem description:

  The SecOC SWS does not make any assumptions or restrictions about dynamic length PDUs. But with parameter SecOCAuthDataFreshnessStartPosition it could be impossible to really use dynamic length in SecOC.

  Additionally in SystemTemplate TPS the constraint constr_3139 talks only about some restrictions when dynamic length IPDUs are used. This would imply dynamic length IPDUs should be possible.

  Without dynamic length support it is also not possible to support securing

complete IDPUM Containers.
One possible solution could be (if dynamic length should not be supported by SecOC) to add a configuration option per IPDUM Container to send always maximum length (padding with 0) to have the static length again.
If dynamic length shall be supported by SecOC a further problem will be the CAN-FD padding.

Agreed solution:

==
AUTOSAR 4.3.1
==

—————-
SRS SecOC
—————-

* Add new section next to 6.1.3.5 and new requirement [SRS_SecOC_xxxx1] Support of padding at lower layer modules and dynamic length Authentic I-PDUs.
* Description: The SecOC module shall be applicable for the use cases with padding at lower layer modules and with dynamic length Authentic I-PDUs.
* Rationale: At receiver side, received Secured I-PDU containing dynamic length Authentic I-PDU may also conatin padding bytes (added by lower layer modules of sender side, to fit to bus-specific L-PDU length constraints, e.g. CAN FD and FlexRay). In such case, receivers cannot identify number of bytes / byte position of the received payload.
* Use Case: dynamic length PDU on CAN FD and FlexRay
* Dependencies: [SRS_SecOC_00012]
* Supporting Material: -

Note: According to CM, RS Main should be referred. But upstream requirements in current (R4.3.0) SRS SecOC are from RS Features, and appropriate requirements are not available in RS Main.
(If we use RS Features, at least [RS_BRF_01568] [RS_BRF_01649] [RS_BRF_01712] [RS_BRF_01716] [RS_BRF_01752] [RS_BRF_02035] [RS_BRF_02036] [RS_BRF_02037] could be mapped to this requirements)

—————-
SWS SecOC
—————-

a1) Adapt Figure 4 in sec. 7.1.1.1

Change from
< (Figure of "Secured I-PDU = Authentic I-PDU | Freshness Value | Authenticator")
< Figure 4: Secured I-PDU contents
to
> (Figure of "Secured I-PDU = Secured I-PDU Header (optional) | Authentic I-PDU | Freshness Value (optional) | Authenticator")
> Figure 4: Secured I-PDU contents
==> to be done as RfC # 77807, not handled in this RfC.

a2) Add new requirements regarding the Secured I-PDU Header and related behavior

Add new requirement [SWS_SecOC_xxxx2] to define the Secured I-PDU Header
> The Secured I-PDU Header shall indicate the length of the Authentic I-PDU in bytes. The length of the Header shall be configurable by the parameter SecOCAuthPduHeaderLength.
> ()
> Note: the SecOC supports combined usage of authentication data in a separate message (secured PDU collection) and Secured I-PDU Header. Also the SecOC covers dynamic length Authentic I-PDU.

Add new requirement [SWS_SecOC_xxxx3] for behavior at transmission (construction) of Secured I-PDUs
> For a Tx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall add the Secured I-PDU Header to the Secured I-PDU with the length of the Authentic I-PDU within the Secured I-PDU, to handle dynamic Authentic I-PDU.
> ()
> Note: Primary purpose of this Header is to indicate the position of Freshness Value and Authenticator in Secured I-PDUs with dynamic length Authentic I-PDU.
> Also some buses which cannot select arbitrary length of L-PDU (e.g. CAN FD and FlexRay) require this Header, because the position of Freshness Value and Authenticator is not always at the end of the Secured I-PDU, as lower layer modules (e.g. CanIf and FrIf) may add bus-specific padding bytes after processing at SecOC (then the L-PDU containing the Secured I-PDU with padding will be: Secured I-PDU = Secured I-PDU Header | Authentic I-PDU | Freshness Value | Authenticator | Bus-specific padding).

Add new requirement [SWS_SecOC_xxxx4] for behavior at reception of Se-

cured I-PDUs
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall assume Secured I-PDU Header shall be available in the Secured I-PDU, to handle dynamic Authentic I-PDU.
> ()

Add new requirement [SWS_SecOC_xxxx5] for behavior at reception of Secured I-PDUs, the Header tells it's longer than the maximum length of the PDU
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0 and the length of Authentic I-PDU in the Header is longer than configured length (in case of dynamic length IPdus (containing a dynamical length signal), this value indicates the maximum data length) of the Authentic I-PDU, the SecOC module shall discard the I-PDU. In such case with SecOC_StartOfReception, BUFREQ_E_NOT_OK shall be returned (see [SWS_COMTYPE_00012]).
> ()
> Note: SecOC_RxIndication has no return value.

Add new requirement [SWS_SecOC_xxxx6] for behavior at reception of Secured I-PDUs, the Header tells it's shorter than received I-PDU length (ignoring the padding at the end)
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall process Secured I-PDU Header, Authentic I-PDU (with the length specified by the Header), Freshness Value and Authenticator of the Rx Secured I-PDU. The rest of bytes in the Secured I-PDU shall be discarded.
> ()

a3) Add a configuration parameter to SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu, SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxAuthenticPdu, SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPdu and SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu to enable/disable Secured I-PDU Header per I-PDU
* SWS Item: ECUC_SecOC_xxxx1
* Name: SecOCAuthPduHeaderLength
* Description:
* This parameter indicates the length (in bytes) of the Secured I-PDU Header in the Secured I-PDU. The length of zero means there's no header in the PDU.
* Multiplicity: 0..1

* Range: 0..4
* Default: 0

a4) (removed)

a8) Update layout definition (construction) for Secured I-PDUs

Change [SWS_SecOC_00037]
from
< [SWS_SecOC_00037]
< The SecOC module shall construct the Secured I-PDU by adding the Freshness Value and the Authenticator to the Authentic I-PDU.
< (SRS_SecOC_00006)
< Note: The Freshness Counter and the Authenticator included as part of the Secured I-PDU may be truncated per configuration specific to the identifier of the Secured I-PDU. The scheme for the Secured I-PDU looks as follows:
< SecuredPDU = AuthenticIPDU | FreshnessValue [SecOCFreshnessValueTxLength] | Authenticator [SecOCAuthInfoTxLength]
to
> [SWS_SecOC_00037]
> The SecOC module shall construct the Secured I-PDU by adding the Secured I-PDU Header (optional), the Freshness Value (optional) and the Authenticator to the Authentic I-PDU.
> The scheme for the Secured I-PDU (includes the order in which the contents are structured in the Secured I-PDU) shall be compliant with below:
> SecuredPDU = SecuredIPDUHeader (optional) | AuthenticIPDU | FreshnessValue [SecOCFreshnessValueTxLength] (optional) | Authenticator [SecOCAuthInfoTxLength]
> (SRS_SecOC_00006)
> Note: The Freshness Counter and the Authenticator included as part of the Secured I-PDU may be truncated per configuration specific to the identifier of the Secured I-PDU. Also Freshness Value may be a part of Authentic I-PDU (see [SWS_SecOC_00219]).
==> to be done as RfC # 77807, not handled in this RfC.

a9) Add new constraints and notes after [SWS_SecOC_00219]:
> [constr_xxxx1] All signals before SecOCAuthDataFreshnessStartPosition within the Secured I-PDU shall have static length.
> Note: SecOC can use a part of the Authentic I-PDU as freshness when SecOCUseAuthDataFreshness=true, only if the part of the Authentic I-PDU to be used as

the freshness is always available at same position in the Authentic I-PDU.

> [constr_xxxx2] Any container I-PDU which contains multiple contained I-PDUs shall be set SecOCUseAuthDataFreshness=false.
> Note: For container PDUs, normally it cannot be ensured which PDU will be put in which position (depends on various timing and trigger conditions). Therefore, container I-PDUs with multiple contained I-PDUs cannot have FV within the Authentic I-PDU.

a10) Adapt Figure 5
==> to be done as RfC # 77807, not handled in this RfC.

a11) Adapt SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxAuthenticPdu and SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu

Change the description of SecOCRxAuthenticPdu (ECUC_SecOC_00061)
from
< This container specifies the Authetic Pdu that is received by the SecOC module from the PduR.
to
> This container specifies the PDU (that is received by the SecOC module from the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.

Change the description of SecOCTxAuthenticPdu ECUC_SecOC_00072
from
< This container specifies the Authetic Pdu that is transmitted by the SecOC module to the PduR after the Mac was generated.
to
> This container specifies the PDU (that is transmitted by the SecOC module to the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.

————————-
TPS System Template (SysT)
————————-

a5) Add new attribute to SecuredIPdu (Table 6.46) which enables SecOCAuthPduHeaderLength>0
* Attribute: useSecuredPduHeader

* Type: SecuredPduHeaderEnum
* Mul.: 0..1
* Kind: attr
* Desc: This attribute defines the size of the header which is inserted into the SecuredIPdu. If this attribute is set to anything but noHeader, the SecuredIPdu contains the Secured I-PDU Header to indicate the length of the AuthenticIPdu. The AuthenticIPdu contains the original payload, i.e. the secured data.

SecuredPduHeaderEnum
- noHeader
- securedPduHeader08Bit
- securedPduHeader16Bit
- securedPduHeader32Bit

Desc: Defines the header which will be inserted into the SecuredIPdu.

a6) Change the description of IPduPort.rxSecurityVerification in Table 6.3: IPduPort:
This attribute defines the bypassing of signature authentication or MAC verification in the receiving ECU.
If not defined or set to true the signature authentication or MAC verification shall be performed for the SecuredIPdu.
If set to false the signature authentication or MAC verification shall not be performed for the SecuredIPdu.

Removed [constr_3139].

TPS_SysT_xxxx2: Setting of useSecuredPduHeader attribute
The useSecuredPduHeader shall be set to a value other than noHeader if the length of the payload Pdu is dynamic and is transmitted over a network which may insert padding bytes depending on the length (e.g. CANFD, Flexray).

Add a note below TPS_SysT_xxxx2:

Please note that the dynamic-length Pdu can be an ISignalIPdu that contains a SystemSignal with dynamicLength set to true. In general it is not possible to run diagnostics on fixed-length Pdus. Therefore, there is a probability that at least a subset of DcmIPdus and UserDefinedIPdus can have dynamic length.

a7) Add upstream mapping between useSecuredPduHeader (SysT) and SecOCAuthPduHeaderLength (EcuC) in C.1.4 SecOc Mapping

b4) Add upstream mapping between rxSecurityVerification (SysT) and SecOCSecuredRxPduVerification (EcuC) in C.1.4 SecOc Mapping

Mapping rule: SecOCSecuredRxPduVerification is True if rxSecurityVerification is not defined, otherwise SecOCSecuredRxPduVerification = rxSecurityVerification

——————-

SRS SecOC

——————-

* Add new section next to 6.1.3.5 (or 6.2.1.1) and new requirement [SRS_SecOC_xxxx2] Support of capability to extract Authentic I-PDU without Authentication
* Description: The SecOC module shall be capable to extract Authentic I-PDU from Secured I-PDU, without Authentication.
* Rationale: SecOC can be used as an extractor of Authentic I-PDU from Secured I-PDU, to enable low latency GW behavior when a part of downstream communication clusters doesn't require authentication of PDUs.
* Use Case: Gateway
* Dependencies: [SRS_SecOC_00025]
* Supporting Material: -

Note: According to CM, RS Main should be referred. But upstream requirments in current (R4.3.0) SRS SecOC are from RS Features, and appropriate requirements are not available in RS Main.
(If we use RS Features, at least [RS_BRF_02035] [RS_BRF_02036] [RS_BRF_02037] could be mapped to this requirements)

——————-

SWS SecOC

——————-

b1) Remove [constr_3139] (not [constr_3193] – sorry, constr_3193 is typo in my comment # 10)

b2) Add new requirements regarding skipped authentication behavior at SecOC (just remove FV/MAC from Secured I-PDU)

* Add new section "Extracting Authentic I-PDU without Authentication at SecOC" or "Skipping Authentication for Secured I-PDUs at SecOC"
* Add new requirement [SWS_SecOC_xxxx7] for behavior of SecOC at reception of Secured I-PDUs without Authentication
> For a Rx Secured I-PDU with SecOCSecuredRxPduVerification=false, the SecOC module shall extract the Authentic I-PDU using the length specified by the Secured

I-PDU Header without Authentication.
> ()

b3) Add a configuration parameter to SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu and SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection to control authentication behavior at SecOC

* SWS Item: ECUC_SecOC_xxxx3
* Name: SecOCSecuredRxPduVerification
* Description: This parameter defines whether the signature authentication or MAC verification shall be performed on this Secured I-PDU. If set to false, the SecOC module extracts the Authentic I-PDU from the Secured I-PDU without verification.
* Multiplicity: 1
* Type: EcucBooleanParamDef
* Default value: false
* Post-Build Variant Value: true
* Value Configuration Class:
* Pre-compile time: X All Variants
* Scope / Dependency: scope: local
–Last change on issue 77336 comment 69–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.18 Specification Item ECUC_SecOC_00093

Trace References:

none

Content:

Name	SecOCAuthPduHeaderLengthSecOCTxSecuredPdu.SecOCAuthPduHeaderLength	
Parent Container	SecOCTxSecuredPdu	
Description	This parameter indicates the length (in bytes) of the Secured I-PDU Header in the Secured I-PDU. The length of zero means there's no header in the PDU.	
Multiplicity	0..1	
Type	EcucIntegerParamDef	
Range	0 .. 4	

Default value	0		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77336: [SECOC] Dynamic length PDUs (Container) not possible / clear

  Problem description:

  The SecOC SWS does not make any assumptions or restrictions about dynamic length PDUs. But with parameter SecOCAuthDataFreshnessStartPosition it could be impossible to really use dynamic length in SecOC.

  Additionally in SystemTemplate TPS the constraint constr_3139 talks only about some restrictions when dynamic length IPDUs are used. This would imply dynamic length IPDUs should be possible.

  Without dynamic length support it is also not possible to support securing complete IDPUM Containers.
  One possible solution could be (if dynamic length should not be supported by SecOC) to add a configuration option per IPDUM Container to send always maximum length (padding with 0) to have the static length again.
  If dynamic length shall be supported by SecOC a further problem will be the CAN-FD padding.

  Agreed solution:

  ==
  AUTOSAR 4.3.1
  ==

  ———————-
  SRS SecOC
  ———————-

  * Add new section next to 6.1.3.5 and new requirement [SRS_SecOC_xxxx1] Support of padding at lower layer modules and dynamic length Authentic I-PDUs.

* Description: The SecOC module shall be applicable for the use cases with padding at lower layer modules and with dynamic length Authentic I-PDUs.
* Rationale: At receiver side, received Secured I-PDU containing dynamic length Authentic I-PDU may also conatin padding bytes (added by lower layer modules of sender side, to fit to bus-specific L-PDU length constraints, e.g. CAN FD and FlexRay). In such case, receivers cannot identify number of bytes / byte position of the received payload.
* Use Case: dynamic length PDU on CAN FD and FlexRay
* Dependencies: [SRS_SecOC_00012]
* Supporting Material: -

Note: According to CM, RS Main should be referred. But upstream requirements in current (R4.3.0) SRS SecOC are from RS Features, and appropriate requirements are not available in RS Main.
(If we use RS Features, at least [RS_BRF_01568] [RS_BRF_01649] [RS_BRF_01712] [RS_BRF_01716] [RS_BRF_01752] [RS_BRF_02035] [RS_BRF_02036] [RS_BRF_02037] could be mapped to this requirements)

——————-
SWS SecOC
——————-

a1) Adapt Figure 4 in sec. 7.1.1.1

Change from
< (Figure of "Secured I-PDU = Authentic I-PDU | Freshness Value | Authenticator")
< Figure 4: Secured I-PDU contents
to
> (Figure of "Secured I-PDU = Secured I-PDU Header (optional) | Authentic I-PDU | Freshness Value (optional) | Authenticator")
> Figure 4: Secured I-PDU contents
==> to be done as RfC # 77807, not handled in this RfC.

a2) Add new requirements regarding the Secured I-PDU Header and related behavior

Add new requirement [SWS_SecOC_xxxx2] to define the Secured I-PDU Header
> The Secured I-PDU Header shall indicate the length of the Authentic I-PDU in bytes. The length of the Header shall be configurable by the parameter SecOCAuthPduHeaderLength.
> ()

> Note: the SecOC supports combined usage of authentication data in a separate message (secured PDU collection) and Secured I-PDU Header. Also the SecOC covers dynamic length Authentic I-PDU.

Add new requirement [SWS_SecOC_xxxx3] for behavior at transmission (construction) of Secured I-PDUs
> For a Tx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall add the Secured I-PDU Header to the Secured I-PDU with the length of the Authentic I-PDU within the Secured I-PDU, to handle dynamic Authentic I-PDU.
> ()
> Note: Primary purpose of this Header is to indicate the position of Freshness Value and Authenticator in Secured I-PDUs with dynamic length Authentic I-PDU.
> Also some buses which cannot select arbitrary length of L-PDU (e.g. CAN FD and FlexRay) require this Header, because the position of Freshness Value and Authenticator is not always at the end of the Secured I-PDU, as lower layer modules (e.g. CanIf and FrIf) may add bus-specific padding bytes after processing at SecOC (then the L-PDU containing the Secured I-PDU with padding will be: Secured I-PDU = Secured I-PDU Header | Authentic I-PDU | Freshness Value | Authenticator | Bus-specific padding).

Add new requirement [SWS_SecOC_xxxx4] for behavior at reception of Secured I-PDUs
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall assume Secured I-PDU Header shall be available in the Secured I-PDU, to handle dynamic Authentic I-PDU.
> ()

Add new requirement [SWS_SecOC_xxxx5] for behavior at reception of Secured I-PDUs, the Header tells it's longer than the maximum length of the PDU
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0 and the length of Authentic I-PDU in the Header is longer than configured length (in case of dynamic length IPdus (containing a dynamical length signal), this value indicates the maximum data length) of the Authentic I-PDU, the SecOC module shall discard the I-PDU. In such case with SecOC_StartOfReception, BUFREQ_E_NOT_OK shall be returned (see [SWS_COMTYPE_00012]).
> ()
> Note: SecOC_RxIndication has no return value.

Add new requirement [SWS_SecOC_xxxx6] for behavior at reception of Secured I-PDUs, the Header tells it's shorter than received I-PDU length (ignoring the padding at the end)
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall process Secured I-PDU Header, Authentic I-PDU (with the length specified by the Header), Freshness Value and Authenticator of the Rx Secured I-PDU. The rest of bytes in the Secured I-PDU shall be discarded.
> ()

a3) Add a configuration parameter to SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu, SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxAuthenticPdu, SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPdu and SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu to enable/disable Secured I-PDU Header per I-PDU
* SWS Item: ECUC_SecOC_xxxx1
* Name: SecOCAuthPduHeaderLength
* Description:
* This parameter indicates the length (in bytes) of the Secured I-PDU Header in the Secured I-PDU. The length of zero means there's no header in the PDU.
* Multiplicity: 0..1
* Range: 0..4
* Default: 0

a4) (removed)

a8) Update layout definition (construction) for Secured I-PDUs

Change [SWS_SecOC_00037]
from
< [SWS_SecOC_00037]
< The SecOC module shall construct the Secured I-PDU by adding the Freshness Value and the Authenticator to the Authentic I-PDU.
< (SRS_SecOC_00006)
< Note: The Freshness Counter and the Authenticator included as part of the Secured I-PDU may be truncated per configuration specific to the identifier of the Secured I-PDU. The scheme for the Secured I-PDU looks as follows:
< SecuredPDU = AuthenticIPDU | FreshnessValue [SecOCFreshnessVal-

ueTxLength] | Authenticator [SecOCAuthInfoTxLength]
to
> [SWS_SecOC_00037]
> The SecOC module shall construct the Secured I-PDU by adding the Secured I-PDU Header (optional), the Freshness Value (optional) and the Authenticator to the Authentic I-PDU.
> The scheme for the Secured I-PDU (includes the order in which the contents are structured in the Secured I-PDU) shall be compliant with below:
> SecuredPDU = SecuredIPDUHeader (optional) | AuthenticIPDU | FreshnessValue [SecOCFreshnessValueTxLength] (optional) | Authenticator [SecOCAuthInfoTxLength]
> (SRS_SecOC_00006)
> Note: The Freshness Counter and the Authenticator included as part of the Secured I-PDU may be truncated per configuration specific to the identifier of the Secured I-PDU. Also Freshness Value may be a part of Authentic I-PDU (see [SWS_SecOC_00219]).
==> to be done as RfC # 77807, not handled in this RfC.

a9) Add new constraints and notes after [SWS_SecOC_00219]:
> [constr_xxxx1] All signals before SecOCAuthDataFreshnessStartPosition within the Secured I-PDU shall have static length.
> Note: SecOC can use a part of the Authentic I-PDU as freshness when SecOCUseAuthDataFreshness=true, only if the part of the Authentic I-PDU to be used as the freshness is always available at same position in the Authentic I-PDU.

> [constr_xxxx2] Any container I-PDU which contains multiple contained I-PDUs shall be set SecOCUseAuthDataFreshness=false.
> Note: For container PDUs, normally it cannot be ensured which PDU will be put in which position (depends on various timing and trigger conditions). Therefore, container I-PDUs with multiple contained I-PDUs cannot have FV within the Authentic I-PDU.

a10) Adapt Figure 5
==> to be done as RfC # 77807, not handled in this RfC.

a11) Adapt SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxAuthenticPdu and SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu

Change the description of SecOCRxAuthenticPdu (ECUC_SecOC_00061)
from
< This container specifies the Authetic Pdu that is received by the SecOC module from the PduR.
to
> This container specifies the PDU (that is received by the SecOC module from the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.

Change the description of SecOCTxAuthenticPdu ECUC_SecOC_00072
from
< This container specifies the Authetic Pdu that is transmitted by the SecOC module to the PduR after the Mac was generated.
to
> This container specifies the PDU (that is transmitted by the SecOC module to the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.

—————-

TPS System Template (SysT)

—————-

a5) Add new attribute to SecuredIPdu (Table 6.46) which enables SecOCAuthPduHeaderLength>0
* Attribute: useSecuredPduHeader
* Type: SecuredPduHeaderEnum
* Mul.: 0..1
* Kind: attr
* Desc: This attribute defines the size of the header which is inserted into the SecuredIPdu. If this attribute is set to anything but noHeader, the SecuredIPdu contains the Secured I-PDU Header to indicate the length of the AuthenticIPdu. The AuthenticIPdu contains the original payload, i.e. the secured data.

SecuredPduHeaderEnum
- noHeader
- securedPduHeader08Bit
- securedPduHeader16Bit
- securedPduHeader32Bit
Desc: Defines the header which will be inserted into the SecuredIPdu.

a6) Change the description of IPduPort.rxSecurityVerification in Table 6.3: IPduPort: This attribute defines the bypassing of signature authentication or MAC verification in the receiving ECU.
If not defined or set to true the signature authentication or MAC verification shall be

performed for the SecuredIPdu.
If set to false the signature authentication or MAC verification shall not be performed for the SecuredIPdu.

Removed [constr_3139].

TPS_SysT_xxxx2: Setting of useSecuredPduHeader attribute
The useSecuredPduHeader shall be set to a value other than noHeader if the length of the payload Pdu is dynamic and is transmitted over a network which may insert padding bytes depending on the length (e.g. CANFD, Flexray).

Add a note below TPS_SysT_xxxx2:

Please note that the dynamic-length Pdu can be an ISignalIPdu that contains a SystemSignal with dynamicLength set to true. In general it is not possible to run diagnostics on fixed-length Pdus. Therefore, there is a probability that at least a subset of DcmIPdus and UserDefinedIPdus can have dynamic length.

a7) Add upstream mapping between useSecuredPduHeader (SysT) and SecOCAuthPduHeaderLength (EcuC) in C.1.4 SecOc Mapping

b4) Add upstream mapping between rxSecurityVerification (SysT) and SecOCSecuredRxPduVerification (EcuC) in C.1.4 SecOc Mapping
Mapping rule: SecOCSecuredRxPduVerification is True if rxSecurityVerification is not defined, otherwise SecOCSecuredRxPduVerification = rxSecurityVerification

———————-
SRS SecOC
———————-

* Add new section next to 6.1.3.5 (or 6.2.1.1) and new requirement [SRS_SecOC_xxxx2] Support of capability to extract Authentic I-PDU without Authentication
* Description: The SecOC module shall be capable to extract Authentic I-PDU from Secured I-PDU, without Authentication.
* Rationale: SecOC can be used as an extractor of Authentic I-PDU from Secured I-PDU, to enable low latency GW behavior when a part of downstream communication clusters doesn't require authentication of PDUs.
* Use Case: Gateway
* Dependencies: [SRS_SecOC_00025]
* Supporting Material: -

Note: According to CM, RS Main should be referred. But upstream requirments in current (R4.3.0) SRS SecOC are from RS Features, and appropriate requirements are not available in RS Main.
(If we use RS Features, at least [RS_BRF_02035] [RS_BRF_02036] [RS_BRF_02037] could be mapped to this requirements)

—————-
SWS SecOC
—————-

b1) Remove [constr_3139] (not [constr_3193] – sorry, constr_3193 is typo in my comment # 10)

b2) Add new requirements regarding skipped authentication behavior at SecOC (just remove FV/MAC from Secured I-PDU)

* Add new section "Extracting Authentic I-PDU without Authentication at SecOC" or "Skipping Authentication for Secured I-PDUs at SecOC"
* Add new requirement [SWS_SecOC_xxxx7] for behavior of SecOC at reception of Secured I-PDUs without Authentication
> For a Rx Secured I-PDU with SecOCSecuredRxPduVerification=false, the SecOC module shall extract the Authentic I-PDU using the length specified by the Secured I-PDU Header without Authentication.
> ()

b3) Add a configuration parameter to SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu and SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection to control authentication behavior at SecOC

* SWS Item: ECUC_SecOC_xxxx3
* Name: SecOCSecuredRxPduVerification
* Description: This parameter defines whether the signature authentication or MAC verification shall be performed on this Secured I-PDU. If set to false, the SecOC module extracts the Authentic I-PDU from the Secured I-PDU without verification.
* Multiplicity: 1
* Type: EcucBooleanParamDef
* Default value: false
* Post-Build Variant Value: true
* Value Configuration Class:
* Pre-compile time: X All Variants

* Scope / Dependency: scope: local
–Last change on issue 77336 comment 69–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.19 Specification Item SWS_SecOC_00034

Trace References:

SRS_SecOC_00006

Content:

The SecOC module shall construct the DataToAuthenticator, i.e. the data that is used to calculate the Authenticator. DataToAuthenticator is formed by concatenating the full 16 bit representation of the Data Id (parameter SecOCDataId), the complete secured part of the Authentic I-PDU and the complete Freshness Value corresponding to SecOCFreshness ValueID in the given order. The Data Id and the Freshness Value shall be encoded in Big Endian byte order for that purpose.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77058: [SecOC] Need assign values in SecOC_VerificationResultType

  Problem description:

  According to [SWS_SecOC_00149], the SecOC_VerificationResultType shall indicate verification results.
  However, current specification does not assign values to members.

  Could you please check and adjust it?

  Agreed solution:

  Add values in [SWS_SecOC_00149] to Range:

  SECOC_VERIFICATIONSUCCESS "0x0"
  SECOC_VERIFICATIONFAILURE "0x1"
  SECOC_FRESHNESSFAILURE "0x2"
  SECOC_AUTHENTICATIONBUILDFAILURE "0x3"

  hint: Please note that RfC 77057 shall be implemented at first.
  –Last change on issue 77058 comment 10–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.20 Specification Item SWS_SecOC_00037

Trace References:

SRS_SecOC_00006

Content:

The SecOC module shall construct the Secured I-PDU by adding the Freshness Value Secured I-PDU Header (optional), the Freshness Value (optional) and the Authenticator to the Authentic I-PDU.

The scheme for the Secured I-PDU (includes the order in which the contents are structured in the Secured I-PDU) shall be compliant with below:

SecuredPDU = SecuredIPDUHeader (optional) | AuthenticIPDU | FreshnessValue [SecOCFreshnessValueTxLength] (optional) | Authenticator [SecOCAuthInfoTxLength]

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77807: [SECOC] Clarify authentication data layout / extend figure 4

  Problem description:

  During WP-A2 Weekly Telco and discussion of RfC # 77336 with WP-M it was identified that the figure 4 states how the layout should look like for secured messages but there is no explicit requirement for that.

  Therefore WP-A2 proposes the following changes:

  - Add a requirement how the layout of the secured I-PDU looks like if authenticated data is transmitted within the Secured I-PDU: At the beginning the (optional) header, directly followed by the authenticated data, directly followed by the authentication information
  - Extend SWS_SecOC_00209 (authenticated data is transmitted separately) regarding layout of the cryptographic I-PDU: At the beginning the (optional) header, directly followed by the authentication information, directly followed by the message linker.
  - Extend figure 4 to have an additional drawing with optional header (currently only one without header exists)

- Add a new figure to have two drawings when secured I-PDU contains only authentication information and data is transported separately - one with header and one without.

This RfC shall be processed after solution for RfC # 77336 has been finished.

Agreed solution:

1) Adapt Figure 4 in sec. 7.1.1.1
Change from
< (Figure of "Secured I-PDU = Authentic I-PDU | Freshness Value | Authenticator")
< Figure 4: Secured I-PDU contents
to
> (Figure of "Secured I-PDU = Secured I-PDU Header (optional) | Authentic I-PDU | Freshness Value (optional) | Authenticator")
> Figure 4: Secured I-PDU contents

2) Update layout definition (construction) for Secured I-PDUs
Change [SWS_SecOC_00037] and its notes
from
< [SWS_SecOC_00037]
< The SecOC module shall construct the Secured I-PDU by adding the Freshness Value and the Authenticator to the Authentic I-PDU.
< (SRS_SecOC_00006)
< Note: The Freshness Counter and the Authenticator included as part of the Secured I-PDU may be truncated per configuration specific to the identifier of the Secured I-PDU. The scheme for the Secured I-PDU looks as follows:
< SecuredPDU = AuthenticIPDU | FreshnessValue [SecOCFreshnessValueTxLength] | Authenticator [SecOCAuthInfoTxLength]
to
> [SWS_SecOC_00037]
> The SecOC module shall construct the Secured I-PDU by adding the Secured I-PDU Header (optional), the Freshness Value (optional) and the Authenticator to the Authentic I-PDU.
> The scheme for the Secured I-PDU (includes the order in which the contents are structured in the Secured I-PDU) shall be compliant with below:
> SecuredPDU = SecuredIPDUHeader (optional) | AuthenticIPDU | FreshnessValue [SecOCFreshnessValueTxLength] (optional) | Authenticator [SecOCAuthInfoTxLength]
> (SRS_SecOC_00006)
> Note: The Freshness Counter and the Authenticator included as part of the Secured I-PDU may be truncated per configuration specific to the identifier of the Secured I-PDU. Also Freshness Value may be a part of Authentic I-PDU (see

[SWS_SecOC_00219]).

3) Adapt text before Figure 5
from
< Figure 5 shows the truncation of the Authenticator and the Freshness Values respecting the parameter SecOCFreshnessValueTxLength and SecOCAuthInfoTxLength.
to
> Figure 5 shows an example of the truncation of the Authenticator and the Freshness Values respecting the parameter SecOCFreshnessValueTxLength and SecOCAuthInfoTxLength.

4) Adapt the title of Figure 5
from
< Figure 5: Secured I-PDU contents with truncated Freshness Counter and truncated Authenticator
to
> Figure 5: An example of Secured I-PDU contents with truncated Freshness Counter and truncated Authenticator (without Secured I-PDU Header)
–Last change on issue 77807 comment 3–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.21 Specification Item SWS_SecOC_00049

Trace References:

SRS_SecOC_00002, SRS_SecOC_00007

Content:

If the verification of a Secured I-PDU was successful, the SecOC module shall pass the information to the Freshness Manager using the Tx-Confirmation functions.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77622: [SecOC] Clarify the "Tx-Confirmation"

  Problem description:

The SWS_SecOC_00049 is mentioned as follows:

"If the verification of a Secured I-PDU was successful, the SecOC module shall pass the information to the Freshness Manager using the Tx-Confirmation functions."

According to our understanding, "Tx-Confirmation" function is not clear because of available some interfaces. Therefore, we suppose to change "Tx-Confirmation" to "SecOC_SPduTxConfirmation".

So, could you please check and correct it?

Agreed solution:

Remove [SWS_SecOC_00049]
Add "If the Freshness Manager requires the status of a secured PDU if it was verified successfully or not, e.g. to synchronize time or counter, then this status shall be taken from the VerificationStatus service provided by SecOC." as a note after [SWS_SecOC_00048]
–Last change on issue 77622 comment 24–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.22 Specification Item SWS_SecOC_00101

Trace References:

SRS_BSW_00337, SRS_BSW_00385, SRS_BSW_00386

Content:

The following errors and exceptions shall be detectable by the SecOC module depending on its build version (development/production mode):

Type or error	Related error code	Value [hex]
An API service was called with a NULL pointer	SECOC_E_PARAM_POINTER	0x01
API service used without module initialization	SECOC_E_UNINIT	0x02
Invalid I-PDU identifier	SECOC_E_INVALID_PDU_SDU_ID	0x03
Crypto service initialization of SecOC failed	SECOCSECCOC_E_CRYPTO_FAILURE INIT_FAILED	0x04 0x07

Type or error	Related error code	Value [hex]
initialization of SecOC Crypto service failed	SECCOCSECOC_E_INIT_FAILED CRYPTO_FAILURE	0x07 0x04

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #59085: Rollout of ’Runtime errors’

  Problem description:

  Inconsistencies in SWS with semantics of Default errors
  –Last change on issue 59085 comment 26–

  Agreed solution:

  solution in Column "G" of the new attachment https://www.autosar.org/bugzilla/attachment.cgi?id=4604

  Notes:
  - It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
  - The review task of the ITs shall be done by the WP to which the specification "belongs".

  *** BSW UML Model ***
  SWS_CanNm:
  ———-
  Chapter 8.6.1 Optional Interfaces:
  Add within SWS_CanNm_00325 the API function Det_ReportRunTimeError

  SWS_LinIf:
  ———-
  SWS_LinIf_00359: add Det_ReportRuntimeError

  SWS_UdpNm:
  ———-
  Replace UDPNM_E_NO_INIT with UDPNM_E_UNINIT in description of API UdpNm_MainFunction_<Instance Id> (SWS_UdpNm_00234)

  *** ECUC XML ***
  Not affected. No configuration of runtime error reporting required (see SWS BSW

General).
–Last change on issue 59085 comment 88–

BW-C-Level:

Application	Specification	Bus
1	4	1

- RfC #76636: Rollout of 'Runtime errors' for entire crypto stack documents

  Problem description:

  Crypto Stack documents are not in line with the RfC # 59085.

  In SWS_secureOnboardCommunication
  Example1: SECOC_E_CRYPTO_FAILURE in the is a development error, but should be a runtime error.

  In SWS_CryptoServiceManager
  Example2: CSM_E_SERVICE_NOT_STARTED is not referenced.

  Example3: CSM_E_PARAM_HANDLE is not referenced in chapter 7.3. It is not clear development error or runtime error.
  –Last change on issue 76636 comment 33–

  Agreed solution:

  CryptoInterface:
  https://bugzilla.autosar.org/attachment.cgi?id=4587

  CryptoServiceManager:
  https://bugzilla.autosar.org/attachment.cgi?id=4614

  CryptoDriver:
  https://bugzilla.autosar.org/attachment.cgi?id=4613

  SecureOnboardCommunication:
  https://bugzilla.autosar.org/attachment.cgi?id=4598
  –Last change on issue 76636 comment 41–

  BW-C-Level:

Application	Specification	Bus
1	4	1

- RfC #77057: [SecOC] Missing definitions in VerificationResultType and DET errors.

Problem description:

According to [SWS_SecOC_00240], the VerificationResultType shall be set to SECOC_AUTHENTICATIONBUILDFAILURE.
However, there are no definitions about SECOC_AUTHENTICATIONBUILDFAILURE in [SWS_SecOC_00149].

Additionally, in [SWS_SecOC_00251], the SecOC module shall report the DET error SECOC_E_FRESHNESS_FAILURE.
However, the SECOC_E_FRESHNESS_FAILURE is missing in [SWS_SecOC_00101].

Could you please check and adjust it?

Agreed solution:

add one definition at the table in [SWS_SecOC_00101]:

SECOC_E_FRESHNESS_FAILURE 0x8

And, add one definition at [SWS_SecOC_00149] to Range:

SECOC_AUTHENTICATIONBUILDFAILURE "Verification not successful because of wrong build authentication codes".

SECOC_AUTHENTICATIONBUILDFAILURE to be added in table SWS_SecOC_00160 (values for verificationStatus).
–Last change on issue 77057 comment 12–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.23 Specification Item SWS_SecOC_00114

Trace References:

SRS_BSW_00337, SRS_BSW_00385, SRS_BSW_00386

Content:

Type or error	Related error code	Value [hex]
NO freshness value available from the Freshness Manager	SECOC_E_FRESHNESS_FAILURE	0x08

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76636: Rollout of 'Runtime errors' for entire crypto stack documents

  Problem description:

  Crypto Stack documents are not in line with the RfC # 59085.

  In SWS_secureOnboardCommunication
  Example1: SECOC_E_CRYPTO_FAILURE in the is a development error, but should be a runtime error.

  In SWS_CryptoServiceManager
  Example2: CSM_E_SERVICE_NOT_STARTED is not referenced.

  Example3: CSM_E_PARAM_HANDLE is not referenced in chapter 7.3. It is not clear development error or runtime error.
  –Last change on issue 76636 comment 33–

  Agreed solution:

  CryptoInterface:
  https://bugzilla.autosar.org/attachment.cgi?id=4587

  CryptoServiceManager:
  https://bugzilla.autosar.org/attachment.cgi?id=4614

  CryptoDriver:
  https://bugzilla.autosar.org/attachment.cgi?id=4613

  SecureOnboardCommunication:
  https://bugzilla.autosar.org/attachment.cgi?id=4598
  –Last change on issue 76636 comment 41–

  BW-C-Level:

Application	Specification	Bus
1	4	1

1.24 Specification Item SWS_SecOC_00128

Trace References:

SRS_BSW_00323, SRS_BSW_00357, SRS_SecOC_00012

Content:

Service name:	SecOC_CopyRxDataSecOC_CopyRxData	
Syntax:	BufReq_ReturnType SecOC_CopyRxData(PduIdType id, const PduInfoType* info, PduLengthType* bufferSizePtr)	
Service ID[hex]:	0x44	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	idSecOC_CopyRxData.id	Identification of the received I-PDU.
	infoSecOC_CopyRxData.info	Provides the source buffer (SduDataPtr) and the number of bytes to be copied (SduLength). An SduLength of 0 can be used to query the current amount of available buffer in the upper layer module. In this case, the SduDataPtr may be a NULL_PTR.
Parameters (inout):	None	
Parameters (out):	bufferSizePtrSecOC_CopyRxData.bufferSizePtr	Available receive buffer after data has been copied.
Return value:	BufReq_ReturnType	BUFREQ_OK: Data copied successfully BUFREQ_E_NOT_OK: Data was not copied because an error occurred.
Description:	This function is called to provide the received data of an I-PDU segment (N-PDU) to the upper layer. Each call to this function provides the next part of the I-PDU data. The size of the remaining data buffer is written to the position indicated by bufferSizePtr.	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77935: [PduR] Misleading description of CopyRxData

  Problem description:

  Name: Martin Schlodder
  Role: Member of WP-A2

  Description/Motivation:

  The description of the CopyRxData API says: "The size of the remaining data is written to the position indicated by bufferSizePtr."

  This text seems to have been copied from the CopyTxData call, where it is correct. CopyRxData should talk about "remaining buffer", not "remaining data".

  Agreed solution:

In the description of the API PduR_<User:LoTp>CopyRxData (SWS_PduR_00512), replace "remaining data" by "remaining buffer".

BW-C-Level:

Application	Specification	Bus
1	1	1

1.25 Specification Item SWS_SecOC_00129

Trace References:

SRS_BSW_00323, SRS_BSW_00357, SRS_SecOC_00012

Content:

Service name:	SecOC_CopyTxDataSecOC_CopyTxData
Syntax:	BufReq_ReturnType SecOC_CopyTxData(PduIdType id, const PduInfoType* info, const RetryInfoType* retry, PduLengthType* availableDataPtr)
Service ID[hex]:	0x43
Sync/Async:	Synchronous
Reentrancy:	Reentrant

Parameters (in):	idSecOC_CopyTxData.id	Identification of the transmitted I-PDU.
	infoSecOC_CopyTxData.info	Provides the destination buffer (SduDataPtr) and the number of bytes to be copied (SduLength). If not enough transmit data is available, no data is copied by the upper layer module and BUFREQ_E_BUSY is returned. The lower layer module may retry the call. An SduLength of 0 can be used to indicate state changes in the retry parameter or to query the current amount of available data in the upper layer module. In this case, the SduDataPtr may be a NULL_PTR.
	retrySecOC_CopyTxData.retry	This parameter is used to acknowledge transmitted data or to retransmit data after transmission problems. If the retry parameter is a NULL_PTR, it indicates that the transmit data can be removed from the buffer immediately after it has been copied. Otherwise, the retry parameter must point to a valid RetryInfoType element. If TpDataState indicates TP_CONFPENDING, the previously copied data must remain in the TP buffer to be available for error recovery. TP_DATACONF indicates that all data that has been copied before this call is confirmed and can be removed from the TP buffer. Data copied by this API call is excluded and will be confirmed later. TP_DATARETRY indicates that this API call shall copy previously copied data in order to recover from an error. In this case TxTpDataCnt specifies the offset in bytes from the current data copy position.
Parameters (inout):	None	
Parameters (out):	availableDataPtrSecOC_CopyTxData.availableDataPtr	Indicates the remaining number of bytes that are available in the upper layer module's Tx buffer. availableDataPtr can be used by TP modules that support dynamic payload lengths (e.g. FrIsoTp) to determine the size of the following CFs.
Return value:	BufReq_ReturnType	BUFREQ_OK: Data has been copied to the transmit buffer completely as requested. BUFREQ_E_BUSY: Request could not be fulfilled, because the required amount of Tx data is not available. The lower layer module may retry this call later on. No data has been copied. BUFREQ_E_NOT_OK: Data has not been copied. Request failed.
Description:	This function is called to acquire the transmit data of an I-PDU segment (N-PDU). Each call to this function provides the next part of the I-PDU data unless retry->TpDataState is TP_DATARETRY. In this case the function restarts to copy the data beginning at the offset from the current position indicated by retry->TxTpDataCnt. The size of the remaining data is written to the position indicated by availableDataPtr.	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #68035: [diverse] Introduce rules defining which input parameters shall be passed per value and which ones per const reference

  Problem description:

  SWS_BSW_00186 especially states that input pointer parameters shall use the const qualifier (i.e., shall be P2CONST).

  In addition to that there shall be a SWS item that states that input parameters of integral and enum type shall be passed by value whereas input parameters of structure type shall be passed by reference.

  The various transformer SWS documents shall be adapted accordingly.
  –Last change on issue 68035 comment 4–

  Agreed solution:

  BSW UML model
  ———–
  The attachment "Changed Proposal in WP-A meeting" contains a list of changes to the APIs in the model (see column H). Afterwards all related documents (included in impact list) shall update their
  generated artifacts.

  General Requirements on Basic Software Modules
  ~~~~~~

  Introduce the following requirements prior to SRS_BSW_00371:

  SRS_BSW_xxxxx: Input parameters of scalar and enum types shall be passed as a value.
  Type: valid
  Description: All input parameters of scalar or enum type shall be passed as a value.
  Rationale:

  Use case: For example a function named <Mip>_SomeFunction with a return type of Std_ReturnType and a single parameter named SomeParameter of type uint8 is defined with the following signature:

  Std_ReturnType <Mip>_SomeFunction(uint8 SomeParameter);
  Dependencies: –
  ~~~~~~

Supporting Material: —

SRS_BSW_yyyyy: Input parameters of structure type shall be passed as a reference to a constant structure
Type: valid
Description: All input parameters of structure type shall be passed as a reference constant structure
Rationale: Passing input parameters of structure type by value would result in additional run-time overhead due to efforts for copying the whole structure.
Use case: For example a function named <Mip>_SomeFunction with a return type of Std_ReturnType and a single parameter named SomeParameter of type SomeStructure (where SomeStructure is a struct) is defined with the following signature:

Std_ReturnType <Mip>_SomeFunction(P2CONST(SomeStructure, AUTOMATIC, <MIP>_APPL_DATA) SomeParameter);
Dependencies: –
Supporting Material: —

SRS_BSW_zzzzz: Input parameters of array type shall be passed as a reference to the constant array base type
Type: valid
Description: All input parameters of array type shall be passed as a reference to the constant array base type
Rationale: This effectively matches the behavior specified in the ISO-C:90 namely that a "declaration of a parameter as ’array of type’ shall be adjusted to ’qualified pointer to
type’".
Use case: For example a function named <Mip>_SomeFunction with a return type of Std_ReturnType and a single parameter named SomeParameter of type array of uint8 is defined with the following signature:

Std_ReturnType <Mip>_SomeFunction(P2CONST(uint8, AUTOMATIC, <MIP>_APPL_DATA) SomeParameter);
Dependencies: –
Supporting Material: —

General Specification of Transformers
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

In SWS_Xfrm_00036 change

const <type>* dataElement

to

<paramtype> dataElement

and add the following to the where clause after the API table after the bullet "type is data type of the data element
"

<paramtype> is derived from <type> according to the parameter passing rules rules defined by the SRS BSW General (see SRS_BSW_xxxxx, SRS_BSW_yyyyy, and SRS_BSW_zzzzz) and SWS BSW General (see SWS_BSW_00186 and SWS_BSW_00187).

In SWS_Xfrm_00038 change

[<type> data_1,] ...
[<type> data_n]

to

[<paramtype> data_1,] ...
[<paramtype> data_n]

and add the following to the where clause after the API table after the bullet "type is data type of the data element
"

<paramtype> is derived from <type> according to the parameter passing rules rules defined by the SRS BSW General (see SRS_BSW_xxxxx, SRS_BSW_yyyyy, and SRS_BSW_zzzzz) and SWS BSW General (see SWS_BSW_00186 and SWS_BSW_00187).

The following paragraph shall then be removed:

For the arguments of ClientServerOperation which are handed over to the transformer as data_1, ..., data_n the requirements to API parameters stated in chapter API Parameters of [5, SWS RTE] are valid (especially [SWS_Rte_01017], [SWS_Rte_01018] and [SWS_Rte_05107]).

In SWS_Xfrm_00040 change

[<originalData1>, ...
<originalDataN>]

to

[<paramtype> originalData1,] ...
[<paramtype> originalDataN]

and add the following to the where clause after the API table after the bullet "type is data type of the data element
"

<paramtype> is derived from <type> according to the parameter passing rules rules defined by the SRS BSW General (see SRS_BSW_xxxxx, SRS_BSW_yyyyy, and SRS_BSW_zzzzz) and SWS BSW General (see SWS_BSW_00186 and SWS_BSW_00187).

In SWS_Xfrm_00044 change

<type> *data_1, ...
<type> *data_n

to

[<paramtype> data_1,] ...
[<paramtype> data_n]

and add the following to the where clause after the API table after the bullet "type is data type of the data element
"

<paramtype> is derived from <type> according to the parameter passing rules rules defined by the SRS BSW General (see SRS_BSW_xxxxx, SRS_BSW_yyyyy, and SRS_BSW_zzzzz) and SWS BSW General (see SWS_BSW_00186 and SWS_BSW_00187).

The following paragraph shall then be removed:

For the arguments of ClientServerOperation which are handed over to the transformer as data_1, ..., data_n the requirements to API parameters stated in chapter API Parameters of [5, SWS RTE] are valid (especially [SWS_Rte_01017], [SWS_Rte_01018] and [SWS_Rte_05107]).

Speci?cation of SOME/IP Transformer
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

In SWS_SomeIpXf_00138 change

const <type>* dataElement

to

<paramtype> dataElement

and add the following to the where clause after the API table after the bullet "type is data type of the data element
"

<paramtype> is derived from <type> according to the parameter passing rules rules defined by the SRS BSW General (see SRS_BSW_xxxxx, SRS_BSW_yyyyy, and SRS_BSW_zzzzz) and SWS BSW General (see SWS_BSW_00186 and SWS_BSW_00187).

In SWS_SomeIpXf_00141 change

[<type> data_1,] ...
[<type> data_n]

to

[<paramtype> data_1,] ...
[<paramtype> data_n]

and add the following to the where clause after the API table after the bullet "type is data type of the data element
"
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

<paramtype> is derived from <type> according to the parameter passing rules rules defined by the SRS BSW General (see SRS_BSW_xxxxx, SRS_BSW_yyyyy, and SRS_BSW_zzzzz) and SWS BSW General (see SWS_BSW_00186 and SWS_BSW_00187).

The following paragraph shall then be removed:

For the arguments of ClientServerOperation which are handed over to the transformer as data_1, ..., data_n the requirements to API parameters stated in chapter API Parameters of [5, SWS RTE] are valid (especially [SWS_Rte_01017], [SWS_Rte_01018] and [SWS_Rte_05107]).

In SWS_SomeIpXf_00145 change

<type> *data_1, ...
<type> *data_n

to

[<paramtype> data_1,] ...
[<paramtype> data_n]

and add the following to the where clause after the API table after the bullet "type is data type of the data element
"

<paramtype> is derived from <type> according to the parameter passing rules rules defined by the SRS BSW General (see SRS_BSW_xxxxx, SRS_BSW_yyyyy, and SRS_BSW_zzzzz) and SWS BSW General (see SWS_BSW_00186 and SWS_BSW_00187).

The following paragraph shall then be removed:

For the arguments of ClientServerOperation which are handed over to the transformer as data_1, ..., data_n the requirements to API parameters stated in chapter API Parameters of [5, SWS RTE] are valid (especially [SWS_Rte_01017], [SWS_Rte_01018] and [SWS_Rte_05107]).

Specification of COM Based Transformer

~~~~~~

In SWS_ComXf_00007 change

const <type>* dataElement

to

<paramtype> dataElement

and add the following to the where clause after the API table after the bullet "type is data type of the data element
"

<paramtype> is derived from <type> according to the parameter passing rules rules defined by the SRS BSW General (see SRS_BSW_xxxxx, SRS_BSW_yyyyy, and SRS_BSW_zzzzz) and SWS BSW General (see SWS_BSW_00186 and SWS_BSW_00187).

Specification of Time Sync over Ethernet
~~~~~~

In SWS_EthTSyn_00040 make the parameter DataPtr of EthTSyn_RxIndication const.

Specification of SWS FlexRay Interface
~~~~~~

Change SWS_FrIf_05073 from
FrIf_NumOfStartupFramesPtr (IN)
to
FrIf_NumOfStartupFramesPtr (OUT)

Specification of ADC
~~~~~~
~[SWS_Adc_00419] Adc_SetupResultBuffer: change Adc_ValueGroupType* to const Adc_ValueGroupType*
~[SWS_Adc_00369] Adc_ReadGroup: move Adc_ValueGroupType * from Parame-

ters (in) to Parameters (out)

There is no need to change parameter from IN to INOUT in Adc_SetupResultBuffer

Specification of Com
~~~~~~
Change type of parameter MetaData of Com_TriggerIPDUSendWithMetaData from uint8* to const uint8*

Specification of ComM
~~~~~~
no change required

Specification of Dem
~~~~~~
no change required

Specification of DLT
~~~~~~
no change required

Specification of DoIP
~~~~~~
From:
Std_ReturnType <User>_DoIPRoutingActivationConfirmation(boolean* Confirmed, uint8* ConfirmationReqData, uint8* ConfirmationResData)
Std_ReturnType <User>_DoIPRoutingActivationAuthentication(boolean* Authenticated, uint8* AuthenticationReqData, uint8* AuthenticationResData)

To:
Std_ReturnType <User>_DoIPRoutingActivationConfirmation(boolean* Confirmed, const uint8* ConfirmationReqData, uint8* ConfirmationResData)
Std_ReturnType <User>_DoIPRoutingActivationAuthentication(boolean* Authenticated, const uint8* AuthenticationReqData, uint8* AuthenticationResData)

Specification of E2ELibrary
~~~~~~

no change required

Specification of Eth
~~~~~~
no change required

Specification of EthIf
~~~~~~
no change required

Specification of EthSwitchDriver
~~~~~~
no change required

Specification of ICUDriver
~~~~~~
SWS_Icu_00201: Icu_StartTimestamp
Parameter (IN): Icu_ValueType* BufferPtr shall be changed to Parameters (out) type

Specification of LdCom
~~~~~~
[SWS_LDCOM_00027]: LdCom_CopyTxData
BufReq_ReturnType LdCom_CopyTxData(PduIdType id, const PduInfoType* info, RetryInfoType* retry, PduLengthType* availableDataPtr) shall be changed to
BufReq_ReturnType LdCom_CopyTxData(PduIdType id, const PduInfoType* info, const RetryInfoType* retry, PduLengthType* availableDataPtr)

[SWS_LDCOM_00036]: Rte_LdComCbkCopyTxData_<sn>
BufReq_ReturnType Rte_LdComCbkCopyTxData_<sn>(const PduInfoType* info, RetryInfoType* retry, PduLengthType* availableDataPtr) shall be changed to
BufReq_ReturnType Rte_LdComCbkCopyTxData_<sn>(const PduInfoType* info, const RetryInfoType* retry, PduLengthType* availableDataPtr)

Specification of Lin
~~~~~~
PduInfoPtr needs to be const in Std_ReturnType Lin_SendFrame(uint8 Channel, const Lin_PduType* PduInfoPtr)

Specification of PduR
~~~~~~
* PduR_<User:LoTp>CopyTxData
add const to "RetryInfoType* retry"

Specification of J1939Nm
~~~~~~
Change parameter ’name’ of User_AddressClaimedIndication to type ’const uint8*’

Specification of SoAd
~~~~~~
=> everything already fixed with RfC 65633

Specification of SPIHandlerDriver
~~~~~~
==> nothing to change for SWS SPI

Specification of SynchronizedTimeBaseManager
~~~~~~
"StbM not affected. All issues listed in the WP-A attachment have been already implemented by IT 69124 in context of RfC 65633"

Specification of TcpIp
~~~~~~
~[SWS_TCPIP_00040] TcpIp_DhcpReadOption: change DataPtr from (IN) to (OUT)
~[SWS_TCPIP_00189] TcpIp_DhcpV6ReadOption: change DataPtr from (IN) to (OUT)
=> everything else already fixed with RfC 65633

Specification of TimeSyncOverFlexRay
~~~~~~
"Change SWS_FrTSyn_00064: parameter versioninfo of type Std_VersionInfoType* is marked wrongly as IN. Change to OUT"
~~~~~~

Specification of EFX
~~~~~~~~~~~~~~~~~~~~~~
~ [SWS_Efx_00355] Efx_Debounce_u8_u8: Include constant for pointer Input-parameter as like below.
uint8 Efx_Debounce_u8_u8(boolean X, Efx_DebounceState_Type * State, const Efx_DebounceParam_Type * Param, sint32 dT)

~ [SWS_Efx_00376] Efx_MedianSort: The parameter <InType>* Array should be InOut instead of In parameter as like below.
Parameters (in): N Size of an array
Parameters (inout): Array Pointer to an array

~ [SWS_Efx_00309] Efx_RampCheckActivity: Include constant for pointer Input-parameter as like below.
boolean Efx_RampCheckActivity(const Efx_StateRamp_Type* State_cpst)

~ [SWS_Efx_00307] Efx_RampGetSwitchPos: Include constant for pointer Input-parameter as like below.
boolean Efx_RampGetSwitchPos(const Efx_StateRamp_Type* State_cpst)

~ [SWS_Efx_00193] Efx_Array_Average: Include constant for pointer Input-parameter as like below.
<OutType> Efx_Array_Average_<InTypeMn>_<OutTypeMn>(const <InType>* Array, uint16 Count)

Specification of MFL
~~~~~~~~~~~~~~~~~~~~~~
~ [SWS_Mfl_00192] Mfl_Debounce_u8_u8: Include constant for pointer Input-parameter as like below.
boolean Mfl_Debounce_u8_u8(boolean X, Mfl_DebounceState_Type* State, const Mfl_DebounceParam_Type* Param, float32 dT)

~ [SWS_Mfl_00266] Mfl_DebounceInit: The parameter Mfl_DebounceState_Type* State should be Out instead of In parameter as like below.
Parameters (in): X Initial value for the input state
Parameters (out): State Pointer to structure for debouncing state variables

~ [SWS_Mfl_00246] Mfl_HystDeltaRight_f32_u8: Include constant for pointer Input-parameter as like below.
boolean Mfl_HystDeltaRight_f32_u8(float32 X, float32 Delta, float32 Rsp, const uint8* State)

~ [SWS_Mfl_00285] Mfl_MedianSort_f32_f32: The parameter Array should be InOut instead of In parameter as like below.
Parameters (in): N Size of an array
Parameters (inout): Array Pointer to an array

~ [SWS_Mfl_00037] Mfl_PT1SetState: The parameter State_cpst should be Out instead of In parameter as like below.
Parameters (in): X1_f32 Initial value for input state
Y1_f32 Initial value for output state
Parameters (out): State_cpst Pointer to internal state structure

~ [SWS_Mfl_00225] Mfl_RampCheckActivity: Include constant for pointer Input-parameter as like below.
boolean Mfl_RampCheckActivity(const Mfl_StateRamp_Type* State_cpst)

~ [SWS_Mfl_00223] Mfl_RampGetSwitchPos: Include constant for pointer Input-parameter as like below.
boolean Mfl_RampGetSwitchPos(const Mfl_StateRamp_Type* State_cpst)
–Last change on issue 68035 comment 135–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.26 Specification Item SWS_SecOC_00137

Trace References:

SRS_BSW_00384

Content:

API function	Description
Det_ReportRuntimeError	Service to report runtime errors. If a callout has been configured then this callout shall be called.
PduR_SecOCCancelTransmit	Requests cancellation of an ongoing transmission of a PDU in a lower layer communication module.
PduR_SecOCIfRxIndication	Indication of a received PDU from a lower layer communication interface module.
PduR_SecOCIfTxConfirmation	The lower layer communication interface module confirms the transmission of a PDU, or the failure to transmit a PDU.
PduR_SecOCTransmit	Requests transmission of a PDU.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #59085: Rollout of ’Runtime errors’

  Problem description:

  Inconsistencies in SWS with semantics of Default errors
  –Last change on issue 59085 comment 26–

  Agreed solution:

  solution in Column "G" of the new attachment https://www.autosar.org/bugzilla/attachment.cgi?id=4604

  Notes:
  - It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
  - The review task of the ITs shall be done by the WP to which the specification "belongs".

  *** BSW UML Model ***
  SWS_CanNm:
  ———-
  Chapter 8.6.1 Optional Interfaces:
  Add within SWS_CanNm_00325 the API function Det_ReportRunTimeError

  SWS_LinIf:
  ———-
  SWS_LinIf_00359: add Det_ReportRuntimeError

  SWS_UdpNm:
  ———-
  Replace UDPNM_E_NO_INIT with UDPNM_E_UNINIT in description of API UdpNm_MainFunction_<Instance Id> (SWS_UdpNm_00234)

  *** ECUC XML ***
  Not affected. No configuration of runtime error reporting required (see SWS BSW General).
  –Last change on issue 59085 comment 88–

  BW-C-Level:

Application	Specification	Bus
1	4	1

1.27 Specification Item SWS_SecOC_00138

Trace References:

SRS_BSW_00384

Content:

API function	Description
Det_ReportError	Service to report development errors.
PduR_SecOCCancelReceive	Requests cancellation of an ongoing reception of a PDU in a lower layer transport protocol module.
PduR_SecOCChangeParameter(obsolete)	Request to change a specific transport protocol parameter (e.g. block size).
PduR_SecOCCopyRxData	This function is called to provide the received data of an I-PDU segment (N-PDU) to the upper layer. Each call to this function provides the next part of the I-PDU data. The size of the remaining data buffer is written to the position indicated by bufferSizePtr.
PduR_SecOCCopyTxData	This function is called to acquire the transmit data of an I-PDU segment (N-PDU). Each call to this function provides the next part of the I-PDU data unless retry->TpDataState is TP_DATARETRY. In this case the function restarts to copy the data beginning at the offset from the current position indicated by retry->TxTpData Cnt. The size of the remaining data is written to the position indicated by availableDataPtr.
PduR_SecOCStartOfReception	This function is called at the start of receiving an N-SDU. The N-SDU might be fragmented into multiple N-PDUs (FF with one or more following CFs) or might consist of a single N-PDU (SF). The service shall provide the currently available maximum buffer size when invoked with TpSduLength equal to 0.
PduR_SecOCTpRxIndication	Called after an I-PDU has been received via the TP API, the result indicates whether the transmission was successful or not.
PduR_SecOCTpTxConfirmation	This function is called after the I-PDU has been transmitted on its network, the result indicates whether the transmission was successful or not.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76404: [Det] Clarifications on runtime errors

  Problem description:

  There are several uncertainties/problems in the SWS DET:
  1. According to SWS_Det_00180, the callouts should have the same signatures

as the corresponding DET functions, but they are void(void) (SWS_Det_00181, SWS_Det_00184, SWS_Det_00187).
2. Section 8.2.3.1 does not describe how the instance ID is passed to DET.
3. Configuration of header files for all three error type callouts are missing.
4. Why does the development error callout reside in DetNotification, while the other two callouts reside in DetGeneral?
5. The limitation in section 4.1 regarding "supervisor mode" does not really make sense. It is assumed that the DET is ignorant regarding the call context, and the software receiving DET callbacks (like DLT or the implementers of the callouts) need to take care of resolving the calling context, if necessary (e.g. in multi-core environments).
6. SWS_Det_00302 defines several runtime errors. But apart from DET_E_CANNOT_REPORT, it is unclear in which situation these errors could be reported by DET: For errors reported by BSW, the DET has no means to validate anything that could lead to such an error. And for SWCs, the modeling already takes care that DET_E_WRONG_MODULE and DET_E_WRONG_INSTANCE cannot occur, while the other two errors can also not be checked by DET without further configuration.
7. Det_ReportTransientFault (SWS_Det_01003) shall return the return value of a configured callout. But what shall happen if more than one callout exisits, and the return different values?
8. SWS_Det_00052: The only API that can result in DET_E_PARAM_POINTER is Det_GetVersionInfo (as the error description mentions correctly). Please reformulate this requirement and move it to section 8.1.3.6 "Det_GetVersionInfo".
–Last change on issue 76404 comment 13–

Agreed solution:

1.
~change SWS_Det_00181/184/187 such that signatures match the APIs
~Figures 3,5, and 7 to be corrected (return missing)
5. remove from 4.1. the sentence: "It is assumed that the whole Basic Software runs in supervisor mode or the switch to supervisor mode is done by a system call within the error reporting function of the DET module."
6. remove SWS_Det_00302 and SWS_Det_00303 and all included errors
7. change SWS_Det_01003 (Return Value-Part only): "Std_ReturnType" If no callout exists it shall return E_OK, otherwise it shall return the value of the configured callout. In case several callouts are configured the logical or (sum) of the callout return values shall be returned. Rationale: since E_OK=0, E_OK will be only returned if all are E_OK, and for multiple error codes there is a good chance to detect several of them.
8. change SWS_Det_00052 from "in case a null pointer error occurs." to "in case a null pointer error occurs in Det_GetVersionInfo." Do not move the requirement, since otherwise the section 7.7 would be empty, but add the following sentence to

8.1.3.6: "In case a null pointer is passed, DET_E_PARAM_POINTER is returned, see SWS_Det_00052."
–Last change on issue 76404 comment 30–

BW-C-Level:

Application	Specification	Bus
1	4	1

- RfC #76491: [PduR] Remove forwarding of ChangeParameter API from PduR

  Problem description:

  The ChangeParameter API is TP specific and there is no use case to call this API through the PduR (CDD or integration code that requires a change in the TP parameter will call the API directly in the TP module)

  Agreed solution:

  PDUR_SWS
  Remove references from ch. 5 "Dependencies to other modules"
  Remove ch. 8.4. "Change transport protocol parameter"

  BSW_Model
  set API PduR_<User:Up>ChangeParameter [SWS_PduR_00482] to obsolete
  set optional interface from [SWS_PduR_00424] to obsolete

  ECUC model
  set [ECUC_PduR_00326] PduRChangeParameterApi to obsolete
  remove reference in description from [ECUC_PduR_00319] PduRUseTag
  –Last change on issue 76491 comment 12–

  BW-C-Level:

Application	Specification	Bus
1	4	1

- RfC #77935: [PduR] Misleading description of CopyRxData

  Problem description:

  Name: Martin Schlodder
  Role: Member of WP-A2

  Description/Motivation:

  The description of the CopyRxData API says: "The size of the remaining data is written to the position indicated by bufferSizePtr."

This text seems to have been copied from the CopyTxData call, where it is correct. CopyRxData should talk about "remaining buffer", not "remaining data".

Agreed solution:

In the description of the API PduR_<User:LoTp>CopyRxData (SWS_PduR_00512), replace "remaining data" by "remaining buffer".

BW-C-Level:

Application	Specification	Bus
1	1	1

1.28 Specification Item SWS_SecOC_00149

Trace References:

SRS_SecOC_00022

Content:

Name	SecOC_VerificationResultTypeSecOC_VerificationResultType		
Kind	Enumeration		
Range	SECOC_VERIFICATIONSUCCESSSecOC_VerificationResultType.SECOC_VERIFICATIONSUCCESS	0x00	Verification successful
	SECOC_VERIFICATIONFAILURESecOC_VerificationResultType.SECOC_VERIFICATIONFAILURE	0x01	Verification not successful
	SECOC_FRESHNESSFAILURESecOC_VerificationResultType.SECOC_FRESHNESSFAILURE	0x02	Verification not successful because of wrong freshness value.
	SECOC_AUTHENTICATIONBUILDFAILURESecOC_VerificationResultType.SECOC_AUTHENTICATIONBUILDFAILURE	0x03	Verification not successful because of wrong build authentication codes
Description	Enumeration to indicate verification results.		
Variation	–		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77057: [SecOC] Missing definitions in VerificationResultType and DET errors.

Problem description:

According to [SWS_SecOC_00240], the VerificationResultType shall be set to SECOC_AUTHENTICATIONBUILDFAILURE.
However, there are no definitions about SECOC_AUTHENTICATIONBUILDFAILURE in [SWS_SecOC_00149].

Additionally, in [SWS_SecOC_00251], the SecOC module shall report the DET error SECOC_E_FRESHNESS_FAILURE.
However, the SECOC_E_FRESHNESS_FAILURE is missing in [SWS_SecOC_00101].

Could you please check and adjust it?

Agreed solution:

add one definition at the table in [SWS_SecOC_00101]:

SECOC_E_FRESHNESS_FAILURE 0x8

And, add one definition at [SWS_SecOC_00149] to Range:

SECOC_AUTHENTICATIONBUILDFAILURE "Verification not successful because of wrong build authentication codes".

SECOC_AUTHENTICATIONBUILDFAILURE to be added in table SWS_SecOC_00160 (values for verificationStatus).
–Last change on issue 77057 comment 12–

BW-C-Level:

Application	**Specification**	**Bus**
1	1	1

- RfC #77058: [SecOC] Need assign values in SecOC_VerificationResultType

  Problem description:

  According to [SWS_SecOC_00149], the SecOC_VerificationResultType shall indicate verification results.
  However, current specification does not assign values to members.

  Could you please check and adjust it?

  Agreed solution:

  Add values in [SWS_SecOC_00149] to Range:

  SECOC_VERIFICATIONSUCCESS "0x0"

SECOC_VERIFICATIONFAILURE "0x1"
SECOC_FRESHNESSFAILURE "0x2"
SECOC_AUTHENTICATIONBUILDFAILURE "0x3"

hint: Please note that RfC 77057 shall be implemented at first.
–Last change on issue 77058 comment 10–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.29 Specification Item SWS_SecOC_00155

Trace References:

SRS_BSW_00385

Content:

If the number of attempts for an Authentic I-PDU has reached the limits of either limit SecOCAuthenticationBuildAttempts or that defines the maximum number of freshness values provided by the freshness manageris reached, the SecOC module shall remove the Authentic I-PDU from its internal buffer and shall report SECOC_E_CRYPTO_FAILURE to the DET module.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77461: [SecOC] How to check the maximum number of freshness values in SecOC module

  Problem description:

  In the SecOC specification, the SWS_SecOC_00155 is mentioned as follows:

  If the number of attempts for an Authentic I-PDU has reached the limits of either SecOCAuthenticationBuildAttempts or the maximum number of freshness values provided by the freshness manager is reached,

  However, we cannot understand how to check the maximum number of freshness values provided by the freshness manager is reached, because the SecOC module don't have the maximum number of freshness value. Therefore, if we need to check the maximum number of freshness values in SecOC module, the SecOC module need to get the maximum number of each freshness value.

I suppose that checking the maximum number of freshness values is not necessary by the SecOC module, because the freshness value is provided from the outer module (the freshness value manager or the Callout function) in R4.3.0.

Could you please check and correct it?

Agreed solution:

Remove the following description:
"or the maximum number of freshness values provided by the freshness manager is reached,"
Change
[SWS_SecOC_00155]
If the number of attempts for an Authentic I-PDU has reached the limits of either SecOCAuthenticationBuildAttempts or the maximum number of freshness values provided by the freshness manager is reached, the SecOC module shall remove the Authentic I-PDU from its internal buffer and shall report SECOC_E_CRYPTO_FAILURE to the DET module.
(SRS_BSW_00385)
into
[SWS_SecOC_00155]
If the number of attempts for an Authentic I-PDU has reached the limit SecOCAuthenticationBuildAttempts that defines the maximum number of freshness values provided by the freshness manager, the SecOC module shall remove the Authentic I-PDU from its internal buffer and shall report SECOC_E_CRYPTO_FAILURE to the DET module.
(SRS_BSW_00385)
–Last change on issue 77461 comment 6–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.30 Specification Item SWS_SecOC_00160

Trace References:

SRS_SecOC_00022

Content:

Name	SecOC_VerificationStatusTypeSecOC_VerificationStatusType
Kind	Structure

Name	SecOC_VerificationStatusTypeSecOC_VerificationStatusType		
Elements	freshnessValueIDSecOC_VerificationStatusType.freshnessValueID	uint16	Identifier of the Freshness Value which resulted in the Verification Status
	verificationStatusSecOC_VerificationStatusType.verificationStatus	SecOC_VerificationResultType	Result of verification attempt: SECOC_VERIFICATIONSUCCESS = Verification successful SECOC_VERIFICATIONFAILURE = Verification not successful SECOC_FRESHNESSFAILURE = Verification not successful because of wrong freshness value SECOC_AUTHENTICATIONBUILDFAILURE = Verification not successful because of wrong build authentication codes
	secOCDataIdSecOC_VerificationStatusType.secOCDataId	uint16	Data ID of SecOCDataId
Description	Data structure to bundle the status of a verification attempt for a specific Freshness Value and Data ID		
Variation	–		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77057: [SecOC] Missing definitions in VerificationResultType and DET errors.

  Problem description:

  According to [SWS_SecOC_00240], the VerificationResultType shall be set to SECOC_AUTHENTICATIONBUILDFAILURE.
  However, there are no definitions about SECOC_AUTHENTICATIONBUILDFAILURE in [SWS_SecOC_00149].

  Additionally, in [SWS_SecOC_00251], the SecOC module shall report the DET error SECOC_E_FRESHNESS_FAILURE.
  However, the SECOC_E_FRESHNESS_FAILURE is missing in [SWS_SecOC_00101].

  Could you please check and adjust it?

  Agreed solution:

  add one definition at the table in [SWS_SecOC_00101]:

  SECOC_E_FRESHNESS_FAILURE 0x8

  And, add one definition at [SWS_SecOC_00149] to Range:

SECOC_AUTHENTICATIONBUILDFAILURE "Verification not successful because of wrong build authentication codes".

SECOC_AUTHENTICATIONBUILDFAILURE to be added in table SWS_SecOC_00160 (values for verificationStatus).
–Last change on issue 77057 comment 12–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.31 Specification Item SWS_SecOC_00194

Trace References:

SRS_SecOC_00003

Content:

This profile depicts one configuration and usage of the JasPar counter base FV with Master-Slave Synchronization method.

It uses the CMAC algorithm based on AES-128 according to NIST SP 800-38B Appendix-A to calculate the MAC. Use the 4 least significant bits of the freshness value as truncated freshness value, and use the 28 most significant bits of the MAC as truncated MAC.

Freshness Value provided to SecOC shall be constructed as described in the [UC_xxx3SecOC_00202]. The profile shall be used for CAN.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76878: [SecOC] Correct "UC_xxx3" to "UC_SecOC_00202"

  Problem description:

  Update incorrect "UC_xxx3" to "UC_SecOC_00202" in [SWS_SecOC_00194].

  Agreed solution:

  Update incorrect "UC_xxx3" to "UC_SecOC_00202" in [SWS_SecOC_00194]
  –Last change on issue 76878 comment 2–

  BW-C-Level:

Application	Specification	Bus
1	1	1

1.32 Specification Item SWS_SecOC_00248

Trace References:

SRS_SecOC_00022, SRS_SecOC_00029

Content:

If the Rx freshness request function returns E_NOT_OK, the verification of an Authentic I-PDU is considered to be failed and the authentication retry counter for this PDU shall be incremented. If the number of authentication attempts has reached SecOCAuthenticationVerifyAttempts, the SecOC module shall remove the Authentic I-PDU from its internal buffer. The failure SECOC_E_RE_FRESHNESS_FAILURE shall be reported to the DET module.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76636: Rollout of 'Runtime errors' for entire crypto stack documents

  Problem description:

  Crypto Stack documents are not in line with the RfC # 59085.

  In SWS_secureOnboardCommunication
  Example1: SECOC_E_CRYPTO_FAILURE in the is a development error, but should be a runtime error.

  In SWS_CryptoServiceManager
  Example2: CSM_E_SERVICE_NOT_STARTED is not referenced.

  Example3: CSM_E_PARAM_HANDLE is not referenced in chapter 7.3. It is not clear development error or runtime error.
  –Last change on issue 76636 comment 33–

  Agreed solution:

  CryptoInterface:
  https://bugzilla.autosar.org/attachment.cgi?id=4587

  CryptoServiceManager:
  https://bugzilla.autosar.org/attachment.cgi?id=4614

  CryptoDriver:
  https://bugzilla.autosar.org/attachment.cgi?id=4613

SecureOnboardCommunication:
https://bugzilla.autosar.org/attachment.cgi?id=4598
–Last change on issue 76636 comment 41–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.33 Specification Item SWS_SecOC_00251

Trace References:

SRS_SecOC_00022, SRS_SecOC_00029

Content:

If DET reporting is enabled via SecOCDevErrorDetect, the SecThe SecOC module shall report the DET error SECOC_E_RE_FRESHNESS_FAILURE when it is finally not able to get the required freshness services for authentication/verification from the Freshness Manager.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76636: Rollout of 'Runtime errors' for entire crypto stack documents

  Problem description:

  Crypto Stack documents are not in line with the RfC # 59085.

  In SWS_secureOnboardCommunication
  Example1: SECOC_E_CRYPTO_FAILURE in the is a development error, but should be a runtime error.

  In SWS_CryptoServiceManager
  Example2: CSM_E_SERVICE_NOT_STARTED is not referenced.

  Example3: CSM_E_PARAM_HANDLE is not referenced in chapter 7.3. It is not clear development error or runtime error.
  –Last change on issue 76636 comment 33–

  Agreed solution:

  CryptoInterface:
  https://bugzilla.autosar.org/attachment.cgi?id=4587

CryptoServiceManager:
https://bugzilla.autosar.org/attachment.cgi?id=4614

CryptoDriver:
https://bugzilla.autosar.org/attachment.cgi?id=4613

SecureOnboardCommunication:
https://bugzilla.autosar.org/attachment.cgi?id=4598
–Last change on issue 76636 comment 41–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.34 Specification Item SWS_SecOC_00261

Trace References:

SRS_SecOC_00006

Content:

The Secured I-PDU Header shall indicate the length of the Authentic I-PDU in bytes. The length of the Header shall be configurable by the parameter SecOCAuthPduHeaderLength.

Note: the SecOC supports combined usage of authentication data in a separate message (secured PDU collection) and Secured I-PDU Header. Also the SecOC covers dynamic length Authentic I-PDU.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77336: [SECOC] Dynamic length PDUs (Container) not possible / clear

  Problem description:

  The SecOC SWS does not make any assumptions or restrictions about dynamic length PDUs. But with parameter SecOCAuthDataFreshnessStartPosition it could be impossible to really use dynamic length in SecOC.

  Additionally in SystemTemplate TPS the constraint constr_3139 talks only about some restrictions when dynamic length IPDUs are used. This would imply dynamic length IPDUs should be possible.

Without dynamic length support it is also not possible to support securing complete IDPUM Containers.
One possible solution could be (if dynamic length should not be supported by SecOC) to add a configuration option per IPDUM Container to send always maximum length (padding with 0) to have the static length again.
If dynamic length shall be supported by SecOC a further problem will be the CAN-FD padding.

Agreed solution:

==
AUTOSAR 4.3.1
==

—————-
SRS SecOC
—————-

* Add new section next to 6.1.3.5 and new requirement [SRS_SecOC_xxxx1] Support of padding at lower layer modules and dynamic length Authentic I-PDUs.
* Description: The SecOC module shall be applicable for the use cases with padding at lower layer modules and with dynamic length Authentic I-PDUs.
* Rationale: At receiver side, received Secured I-PDU containing dynamic length Authentic I-PDU may also conatin padding bytes (added by lower layer modules of sender side, to fit to bus-specific L-PDU length constraints, e.g. CAN FD and FlexRay). In such case, receivers cannot identify number of bytes / byte position of the received payload.
* Use Case: dynamic length PDU on CAN FD and FlexRay
* Dependencies: [SRS_SecOC_00012]
* Supporting Material: -

Note: According to CM, RS Main should be referred. But upstream requirements in current (R4.3.0) SRS SecOC are from RS Features, and appropriate requirements are not available in RS Main.
(If we use RS Features, at least [RS_BRF_01568] [RS_BRF_01649] [RS_BRF_01712] [RS_BRF_01716] [RS_BRF_01752] [RS_BRF_02035] [RS_BRF_02036] [RS_BRF_02037] could be mapped to this requirements)

—————-
SWS SecOC
—————-

a1) Adapt Figure 4 in sec. 7.1.1.1

Change from
< (Figure of "Secured I-PDU = Authentic I-PDU | Freshness Value | Authenticator")
< Figure 4: Secured I-PDU contents
to
> (Figure of "Secured I-PDU = Secured I-PDU Header (optional) | Authentic I-PDU | Freshness Value (optional) | Authenticator")
> Figure 4: Secured I-PDU contents
==> to be done as RfC # 77807, not handled in this RfC.

a2) Add new requirements regarding the Secured I-PDU Header and related behavior

Add new requirement [SWS_SecOC_xxxx2] to define the Secured I-PDU Header
> The Secured I-PDU Header shall indicate the length of the Authentic I-PDU in bytes. The length of the Header shall be configurable by the parameter SecOCAuthPduHeaderLength.
> ()
> Note: the SecOC supports combined usage of authentication data in a separate message (secured PDU collection) and Secured I-PDU Header. Also the SecOC covers dynamic length Authentic I-PDU.

Add new requirement [SWS_SecOC_xxxx3] for behavior at transmission (construction) of Secured I-PDUs
> For a Tx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall add the Secured I-PDU Header to the Secured I-PDU with the length of the Authentic I-PDU within the Secured I-PDU, to handle dynamic Authentic I-PDU.
> ()
> Note: Primary purpose of this Header is to indicate the position of Freshness Value and Authenticator in Secured I-PDUs with dynamic length Authentic I-PDU.
> Also some buses which cannot select arbitrary length of L-PDU (e.g. CAN FD and FlexRay) require this Header, because the position of Freshness Value and Authenticator is not always at the end of the Secured I-PDU, as lower layer modules (e.g. CanIf and FrIf) may add bus-specific padding bytes after processing at SecOC (then the L-PDU containing the Secured I-PDU with padding will be: Secured I-PDU = Secured I-PDU Header | Authentic I-PDU | Freshness Value | Authenticator | Bus-specific padding).

Add new requirement [SWS_SecOC_xxxx4] for behavior at reception of Secured I-PDUs
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall assume Secured I-PDU Header shall be available in the Secured I-PDU, to handle dynamic Authentic I-PDU.
> ()

Add new requirement [SWS_SecOC_xxxx5] for behavior at reception of Secured I-PDUs, the Header tells it's longer than the maximum length of the PDU
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0 and the length of Authentic I-PDU in the Header is longer than configured length (in case of dynamic length IPdus (containing a dynamical length signal), this value indicates the maximum data length) of the Authentic I-PDU, the SecOC module shall discard the I-PDU. In such case with SecOC_StartOfReception, BUFREQ_E_NOT_OK shall be returned (see [SWS_COMTYPE_00012]).
> ()
> Note: SecOC_RxIndication has no return value.

Add new requirement [SWS_SecOC_xxxx6] for behavior at reception of Secured I-PDUs, the Header tells it's shorter than received I-PDU length (ignoring the padding at the end)
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall process Secured I-PDU Header, Authentic I-PDU (with the length specified by the Header), Freshness Value and Authenticator of the Rx Secured I-PDU. The rest of bytes in the Secured I-PDU shall be discarded.
> ()

a3) Add a configuration parameter to SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu, SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxAuthenticPdu, SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPdu and SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu to enable/disable Secured I-PDU Header per I-PDU
* SWS Item: ECUC_SecOC_xxxx1
* Name: SecOCAuthPduHeaderLength
* Description:
* This parameter indicates the length (in bytes) of the Secured I-PDU Header in the

Secured I-PDU. The length of zero means there's no header in the PDU.
* Multiplicity: 0..1
* Range: 0..4
* Default: 0

a4) (removed)

a8) Update layout definition (construction) for Secured I-PDUs

Change [SWS_SecOC_00037]
from
< [SWS_SecOC_00037]
< The SecOC module shall construct the Secured I-PDU by adding the Freshness Value and the Authenticator to the Authentic I-PDU.
< (SRS_SecOC_00006)
< Note: The Freshness Counter and the Authenticator included as part of the Secured I-PDU may be truncated per configuration specific to the identifier of the Secured I-PDU. The scheme for the Secured I-PDU looks as follows:
< SecuredPDU = AuthenticIPDU | FreshnessValue [SecOCFreshnessValueTxLength] | Authenticator [SecOCAuthInfoTxLength]
to
> [SWS_SecOC_00037]
> The SecOC module shall construct the Secured I-PDU by adding the Secured I-PDU Header (optional), the Freshness Value (optional) and the Authenticator to the Authentic I-PDU.
> The scheme for the Secured I-PDU (includes the order in which the contents are structured in the Secured I-PDU) shall be compliant with below:
> SecuredPDU = SecuredIPDUHeader (optional) | AuthenticIPDU | FreshnessValue [SecOCFreshnessValueTxLength] (optional) | Authenticator [SecOCAuthInfoTxLength]
> (SRS_SecOC_00006)
> Note: The Freshness Counter and the Authenticator included as part of the Secured I-PDU may be truncated per configuration specific to the identifier of the Secured I-PDU. Also Freshness Value may be a part of Authentic I-PDU (see [SWS_SecOC_00219]).
==> to be done as RfC # 77807, not handled in this RfC.

a9) Add new constraints and notes after [SWS_SecOC_00219]:
> [constr_xxxx1] All signals before SecOCAuthDataFreshnessStartPosition within the Secured I-PDU shall have static length.

> Note: SecOC can use a part of the Authentic I-PDU as freshness when SecOCUseAuthDataFreshness=true, only if the part of the Authentic I-PDU to be used as the freshness is always available at same position in the Authentic I-PDU.

> [constr_xxxx2] Any container I-PDU which contains multiple contained I-PDUs shall be set SecOCUseAuthDataFreshness=false.
> Note: For container PDUs, normally it cannot be ensured which PDU will be put in which position (depends on various timing and trigger conditions). Therefore, container I-PDUs with multiple contained I-PDUs cannot have FV within the Authentic I-PDU.

a10) Adapt Figure 5
==> to be done as RfC # 77807, not handled in this RfC.

a11) Adapt SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxAuthenticPdu and SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu

Change the description of SecOCRxAuthenticPdu (ECUC_SecOC_00061)
from
< This container specifies the Authetic Pdu that is received by the SecOC module from the PduR.
to
> This container specifies the PDU (that is received by the SecOC module from the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.

Change the description of SecOCTxAuthenticPdu ECUC_SecOC_00072
from
< This container specifies the Authetic Pdu that is transmitted by the SecOC module to the PduR after the Mac was generated.
to
> This container specifies the PDU (that is transmitted by the SecOC module to the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.

—————-

TPS System Template (SysT)

—————-

a5) Add new attribute to SecuredIPdu (Table 6.46) which enables SecOCAu-

thPduHeaderLength>0
* Attribute: useSecuredPduHeader
* Type: SecuredPduHeaderEnum
* Mul.: 0..1
* Kind: attr
* Desc: This attribute defines the size of the header which is inserted into the SecuredIPdu. If this attribute is set to anything but noHeader, the SecuredIPdu contains the Secured I-PDU Header to indicate the length of the AuthenticIPdu. The AuthenticIPdu contains the original payload, i.e. the secured data.

SecuredPduHeaderEnum
- noHeader
- securedPduHeader08Bit
- securedPduHeader16Bit
- securedPduHeader32Bit
Desc: Defines the header which will be inserted into the SecuredIPdu.

a6) Change the description of IPduPort.rxSecurityVerification in Table 6.3: IPduPort: This attribute defines the bypassing of signature authentication or MAC verification in the receiving ECU.
If not defined or set to true the signature authentication or MAC verification shall be performed for the SecuredIPdu.
If set to false the signature authentication or MAC verification shall not be performed for the SecuredIPdu.

Removed [constr_3139].

TPS_SysT_xxxx2: Setting of useSecuredPduHeader attribute
The useSecuredPduHeader shall be set to a value other than noHeader if the length of the payload Pdu is dynamic and is transmitted over a network which may insert padding bytes depending on the length (e.g. CANFD, Flexray).

Add a note below TPS_SysT_xxxx2:

Please note that the dynamic-length Pdu can be an ISignalIPdu that contains a SystemSignal with dynamicLength set to true. In general it is not possible to run diagnostics on fixed-length Pdus. Therefore, there is a probability that at least a subset of DcmIPdus and UserDefinedIPdus can have dynamic length.

a7) Add upstream mapping between useSecuredPduHeader (SysT) and SecOCAuthPduHeaderLength (EcuC) in C.1.4 SecOc Mapping

b4) Add upstream mapping between rxSecurityVerification (SysT) and SecOCSecuredRxPduVerification (EcuC) in C.1.4 SecOc Mapping
Mapping rule: SecOCSecuredRxPduVerification is True if rxSecurityVerification is not defined, otherwise SecOCSecuredRxPduVerification = rxSecurityVerification

—————-

SRS SecOC

—————-

* Add new section next to 6.1.3.5 (or 6.2.1.1) and new requirement [SRS_SecOC_xxxx2] Support of capability to extract Authentic I-PDU without Authentication
* Description: The SecOC module shall be capable to extract Authentic I-PDU from Secured I-PDU, without Authentication.
* Rationale: SecOC can be used as an extractor of Authentic I-PDU from Secured I-PDU, to enable low latency GW behavior when a part of downstream communication clusters doesn't require authentication of PDUs.
* Use Case: Gateway
* Dependencies: [SRS_SecOC_00025]
* Supporting Material: -

Note: According to CM, RS Main should be referred. But upstream requirments in current (R4.3.0) SRS SecOC are from RS Features, and appropriate requirements are not available in RS Main.
(If we use RS Features, at least [RS_BRF_02035] [RS_BRF_02036] [RS_BRF_02037] could be mapped to this requirements)

—————-

SWS SecOC

—————-

b1) Remove [constr_3139] (not [constr_3193] – sorry, constr_3193 is typo in my comment # 10)

b2) Add new requirements regarding skipped authentication behavior at SecOC (just remove FV/MAC from Secured I-PDU)

* Add new section "Extracting Authentic I-PDU without Authentication at SecOC" or "Skipping Authentication for Secured I-PDUs at SecOC"
* Add new requirement [SWS_SecOC_xxxx7] for behavior of SecOC at reception of Secured I-PDUs without Authentication

> For a Rx Secured I-PDU with SecOCSecuredRxPduVerification=false, the SecOC module shall extract the Authentic I-PDU using the length specified by the Secured I-PDU Header without Authentication.
> ()

b3) Add a configuration parameter to SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu and SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection to control authentication behavior at SecOC

* SWS Item: ECUC_SecOC_xxxx3
* Name: SecOCSecuredRxPduVerification
* Description: This parameter defines whether the signature authentication or MAC verification shall be performed on this Secured I-PDU. If set to false, the SecOC module extracts the Authentic I-PDU from the Secured I-PDU without verification.
* Multiplicity: 1
* Type: EcucBooleanParamDef
* Default value: false
* Post-Build Variant Value: true
* Value Configuration Class:
* Pre-compile time: X All Variants
* Scope / Dependency: scope: local
–Last change on issue 77336 comment 69–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.35 Specification Item SWS_SecOC_00262

Trace References:

SRS_SecOC_00006

Content:

For a Tx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall add the Secured I-PDU Header to the Secured I-PDU with the length of the Authentic I-PDU within the Secured I-PDU, to handle dynamic Authentic I-PDU.

Note: Primary purpose of this Header is to indicate the position of Freshness Value and Authenticator in Secured I-PDUs with dynamic length Authentic I-PDU. Also some buses which cannot select arbitrary length of L-PDU (e.g. CAN FD and FlexRay) require this

Header, because the position of Freshness Value and Authenticator is not always at the end of the Secured I-PDU, as lower layer modules (e.g. CanIf and FrIf) may add bus-specific padding bytes after processing at SecOC (then the L-PDU containing the Secured I-PDU with padding will be: Secured I-PDU = Secured I-PDU Header | Authentic I-PDU | Freshness Value | Authenticator | Bus-specific padding).

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77336: [SECOC] Dynamic length PDUs (Container) not possible / clear

  Problem description:

  The SecOC SWS does not make any assumptions or restrictions about dynamic length PDUs. But with parameter SecOCAuthDataFreshnessStartPosition it could be impossible to really use dynamic length in SecOC.

  Additionally in SystemTemplate TPS the constraint constr_3139 talks only about some restrictions when dynamic length IPDUs are used. This would imply dynamic length IPDUs should be possible.

  Without dynamic length support it is also not possible to support securing complete IDPUM Containers.
  One possible solution could be (if dynamic length should not be supported by SecOC) to add a configuration option per IPDUM Container to send always maximum length (padding with 0) to have the static length again.
  If dynamic length shall be supported by SecOC a further problem will be the CAN-FD padding.

  Agreed solution:

  ==
  AUTOSAR 4.3.1
  ==

  ————————-
  SRS SecOC
  ————————-

  * Add new section next to 6.1.3.5 and new requirement [SRS_SecOC_xxxx1] Support of padding at lower layer modules and dynamic length Authentic I-PDUs.
  * Description: The SecOC module shall be applicable for the use cases with padding at lower layer modules and with dynamic length Authentic I-PDUs.
  * Rationale: At receiver side, received Secured I-PDU containing dynamic length Authentic I-PDU may also conatin padding bytes (added by lower layer modules

of sender side, to fit to bus-specific L-PDU length constraints, e.g. CAN FD and FlexRay). In such case, receivers cannot identify number of bytes / byte position of the received payload.
* Use Case: dynamic length PDU on CAN FD and FlexRay
* Dependencies: [SRS_SecOC_00012]
* Supporting Material: -

Note: According to CM, RS Main should be referred. But upstream requirements in current (R4.3.0) SRS SecOC are from RS Features, and appropriate requirements are not available in RS Main.
(If we use RS Features, at least [RS_BRF_01568] [RS_BRF_01649] [RS_BRF_01712] [RS_BRF_01716] [RS_BRF_01752] [RS_BRF_02035] [RS_BRF_02036] [RS_BRF_02037] could be mapped to this requirements)

—————-
SWS SecOC
—————-

a1) Adapt Figure 4 in sec. 7.1.1.1

Change from
< (Figure of "Secured I-PDU = Authentic I-PDU | Freshness Value | Authenticator")
< Figure 4: Secured I-PDU contents
to
> (Figure of "Secured I-PDU = Secured I-PDU Header (optional) | Authentic I-PDU | Freshness Value (optional) | Authenticator")
> Figure 4: Secured I-PDU contents
==> to be done as RfC # 77807, not handled in this RfC.

a2) Add new requirements regarding the Secured I-PDU Header and related behavior

Add new requirement [SWS_SecOC_xxxx2] to define the Secured I-PDU Header
> The Secured I-PDU Header shall indicate the length of the Authentic I-PDU in bytes. The length of the Header shall be configurable by the parameter SecOCAuthPduHeaderLength.
> ()
> Note: the SecOC supports combined usage of authentication data in a separate message (secured PDU collection) and Secured I-PDU Header. Also the SecOC covers dynamic length Authentic I-PDU.

Add new requirement [SWS_SecOC_xxxx3] for behavior at transmission (construction) of Secured I-PDUs
> For a Tx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall add the Secured I-PDU Header to the Secured I-PDU with the length of the Authentic I-PDU within the Secured I-PDU, to handle dynamic Authentic I-PDU.
> ()
> Note: Primary purpose of this Header is to indicate the position of Freshness Value and Authenticator in Secured I-PDUs with dynamic length Authentic I-PDU.
> Also some buses which cannot select arbitrary length of L-PDU (e.g. CAN FD and FlexRay) require this Header, because the position of Freshness Value and Authenticator is not always at the end of the Secured I-PDU, as lower layer modules (e.g. CanIf and FrIf) may add bus-specific padding bytes after processing at SecOC (then the L-PDU containing the Secured I-PDU with padding will be: Secured I-PDU = Secured I-PDU Header | Authentic I-PDU | Freshness Value | Authenticator | Bus-specific padding).

Add new requirement [SWS_SecOC_xxxx4] for behavior at reception of Secured I-PDUs
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall assume Secured I-PDU Header shall be available in the Secured I-PDU, to handle dynamic Authentic I-PDU.
> ()

Add new requirement [SWS_SecOC_xxxx5] for behavior at reception of Secured I-PDUs, the Header tells it's longer than the maximum length of the PDU
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0 and the length of Authentic I-PDU in the Header is longer than configured length (in case of dynamic length IPdus (containing a dynamical length signal), this value indicates the maximum data length) of the Authentic I-PDU, the SecOC module shall discard the I-PDU. In such case with SecOC_StartOfReception, BUFREQ_E_NOT_OK shall be returned (see [SWS_COMTYPE_00012]).
> ()
> Note: SecOC_RxIndication has no return value.

Add new requirement [SWS_SecOC_xxxx6] for behavior at reception of Secured I-PDUs, the Header tells it's shorter than received I-PDU length (ignoring the padding at the end)
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC

module shall process Secured I-PDU Header, Authentic I-PDU (with the length specified by the Header), Freshness Value and Authenticator of the Rx Secured I-PDU. The rest of bytes in the Secured I-PDU shall be discarded.
> ()

a3) Add a configuration parameter to SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu, SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxAuthenticPdu, SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPdu and SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu to enable/disable Secured I-PDU Header per I-PDU
* SWS Item: ECUC_SecOC_xxxx1
* Name: SecOCAuthPduHeaderLength
* Description:
* This parameter indicates the length (in bytes) of the Secured I-PDU Header in the Secured I-PDU. The length of zero means there's no header in the PDU.
* Multiplicity: 0..1
* Range: 0..4
* Default: 0

a4) (removed)

a8) Update layout definition (construction) for Secured I-PDUs

Change [SWS_SecOC_00037]
from
< [SWS_SecOC_00037]
< The SecOC module shall construct the Secured I-PDU by adding the Freshness Value and the Authenticator to the Authentic I-PDU.
< (SRS_SecOC_00006)
< Note: The Freshness Counter and the Authenticator included as part of the Secured I-PDU may be truncated per configuration specific to the identifier of the Secured I-PDU. The scheme for the Secured I-PDU looks as follows:
< SecuredPDU = AuthenticIPDU | FreshnessValue [SecOCFreshnessValueTxLength] | Authenticator [SecOCAuthInfoTxLength]
to
> [SWS_SecOC_00037]
> The SecOC module shall construct the Secured I-PDU by adding the Secured

I-PDU Header (optional), the Freshness Value (optional) and the Authenticator to the Authentic I-PDU.
> The scheme for the Secured I-PDU (includes the order in which the contents are structured in the Secured I-PDU) shall be compliant with below:
> SecuredPDU = SecuredIPDUHeader (optional) | AuthenticIPDU | FreshnessValue [SecOCFreshnessValueTxLength] (optional) | Authenticator [SecOCAuthInfoTxLength]
> (SRS_SecOC_00006)
> Note: The Freshness Counter and the Authenticator included as part of the Secured I-PDU may be truncated per configuration specific to the identifier of the Secured I-PDU. Also Freshness Value may be a part of Authentic I-PDU (see [SWS_SecOC_00219]).
==> to be done as RfC # 77807, not handled in this RfC.

a9) Add new constraints and notes after [SWS_SecOC_00219]:
> [constr_xxxx1] All signals before SecOCAuthDataFreshnessStartPosition within the Secured I-PDU shall have static length.
> Note: SecOC can use a part of the Authentic I-PDU as freshness when SecOCUseAuthDataFreshness=true, only if the part of the Authentic I-PDU to be used as the freshness is always available at same position in the Authentic I-PDU.

> [constr_xxxx2] Any container I-PDU which contains multiple contained I-PDUs shall be set SecOCUseAuthDataFreshness=false.
> Note: For container PDUs, normally it cannot be ensured which PDU will be put in which position (depends on various timing and trigger conditions). Therefore, container I-PDUs with multiple contained I-PDUs cannot have FV within the Authentic I-PDU.

a10) Adapt Figure 5
==> to be done as RfC # 77807, not handled in this RfC.

a11) Adapt SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxAuthenticPdu and SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu

Change the description of SecOCRxAuthenticPdu (ECUC_SecOC_00061)
from
< This container specifies the Authetic Pdu that is received by the SecOC module from the PduR.

to
> This container specifies the PDU (that is received by the SecOC module from the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.

Change the description of SecOCTxAuthenticPdu ECUC_SecOC_00072
from
< This container specifies the Authetic Pdu that is transmitted by the SecOC module to the PduR after the Mac was generated.
to
> This container specifies the PDU (that is transmitted by the SecOC module to the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.

—————-
TPS System Template (SysT)
—————-

a5) Add new attribute to SecuredIPdu (Table 6.46) which enables SecOCAuthPduHeaderLength>0
* Attribute: useSecuredPduHeader
* Type: SecuredPduHeaderEnum
* Mul.: 0..1
* Kind: attr
* Desc: This attribute defines the size of the header which is inserted into the SecuredIPdu. If this attribute is set to anything but noHeader, the SecuredIPdu contains the Secured I-PDU Header to indicate the length of the AuthenticIPdu. The AuthenticIPdu contains the original payload, i.e. the secured data.

SecuredPduHeaderEnum
- noHeader
- securedPduHeader08Bit
- securedPduHeader16Bit
- securedPduHeader32Bit
Desc: Defines the header which will be inserted into the SecuredIPdu.

a6) Change the description of IPduPort.rxSecurityVerification in Table 6.3: IPduPort:
This attribute defines the bypassing of signature authentication or MAC verification in the receiving ECU.
If not defined or set to true the signature authentication or MAC verification shall be performed for the SecuredIPdu.
If set to false the signature authentication or MAC verification shall not be performed for the SecuredIPdu.

Removed [constr_3139].

TPS_SysT_xxxx2: Setting of useSecuredPduHeader attribute
The useSecuredPduHeader shall be set to a value other than noHeader if the length of the payload Pdu is dynamic and is transmitted over a network which may insert padding bytes depending on the length (e.g. CANFD, Flexray).

Add a note below TPS_SysT_xxxx2:

Please note that the dynamic-length Pdu can be an ISignalIPdu that contains a SystemSignal with dynamicLength set to true. In general it is not possible to run diagnostics on fixed-length Pdus. Therefore, there is a probability that at least a subset of DcmIPdus and UserDefinedIPdus can have dynamic length.

a7) Add upstream mapping between useSecuredPduHeader (SysT) and SecOCAuthPduHeaderLength (EcuC) in C.1.4 SecOc Mapping

b4) Add upstream mapping between rxSecurityVerification (SysT) and SecOCSecuredRxPduVerification (EcuC) in C.1.4 SecOc Mapping
Mapping rule: SecOCSecuredRxPduVerification is True if rxSecurityVerification is not defined, otherwise SecOCSecuredRxPduVerification = rxSecurityVerification

—————-

SRS SecOC

—————-

* Add new section next to 6.1.3.5 (or 6.2.1.1) and new requirement [SRS_SecOC_xxxx2] Support of capability to extract Authentic I-PDU without Authentication
* Description: The SecOC module shall be capable to extract Authentic I-PDU from Secured I-PDU, without Authentication.
* Rationale: SecOC can be used as an extractor of Authentic I-PDU from Secured I-PDU, to enable low latency GW behavior when a part of downstream communication clusters doesn't require authentication of PDUs.
* Use Case: Gateway
* Dependencies: [SRS_SecOC_00025]
* Supporting Material: -

Note: According to CM, RS Main should be referred. But upstream requirments in current (R4.3.0) SRS SecOC are from RS Features, and appropriate requirements are not available in RS Main.
(If we use RS Features, at least [RS_BRF_02035] [RS_BRF_02036]

[RS_BRF_02037] could be mapped to this requirements)

———————-

SWS SecOC

———————-

b1) Remove [constr_3139] (not [constr_3193] – sorry, constr_3193 is typo in my comment # 10)

b2) Add new requirements regarding skipped authentication behavior at SecOC (just remove FV/MAC from Secured I-PDU)

* Add new section "Extracting Authentic I-PDU without Authentication at SecOC" or "Skipping Authentication for Secured I-PDUs at SecOC"
* Add new requirement [SWS_SecOC_xxxx7] for behavior of SecOC at reception of Secured I-PDUs without Authentication
> For a Rx Secured I-PDU with SecOCSecuredRxPduVerification=false, the SecOC module shall extract the Authentic I-PDU using the length specified by the Secured I-PDU Header without Authentication.
> ()

b3) Add a configuration parameter to SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu and SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection to control authentication behavior at SecOC

* SWS Item: ECUC_SecOC_xxxx3
* Name: SecOCSecuredRxPduVerification
* Description: This parameter defines whether the signature authentication or MAC verification shall be performed on this Secured I-PDU. If set to false, the SecOC module extracts the Authentic I-PDU from the Secured I-PDU without verification.
* Multiplicity: 1
* Type: EcucBooleanParamDef
* Default value: false
* Post-Build Variant Value: true
* Value Configuration Class:
* Pre-compile time: X All Variants
* Scope / Dependency: scope: local
–Last change on issue 77336 comment 69–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.36 Specification Item SWS_SecOC_00263

Trace References:

SRS_BSW_00385

Content:

For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0 and the length of Authentic I-PDU in the Header is longer than configured length (in case of dynamic length IPdus (containing a dynamical length signal), this value indicates the maximum data length) of the Authentic I-PDU, the SecOC module shall discard the I-PDU. In such case with SecOC_StartOfReception, BUFREQ_E_NOT_OK shall be returned (see [SWS_COMTYPE_00012]).

Note: SecOC_RxIndication has no return value.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77336: [SECOC] Dynamic length PDUs (Container) not possible / clear

  Problem description:

  The SecOC SWS does not make any assumptions or restrictions about dynamic length PDUs. But with parameter SecOCAuthDataFreshnessStartPosition it could be impossible to really use dynamic length in SecOC.

  Additionally in SystemTemplate TPS the constraint constr_3139 talks only about some restrictions when dynamic length IPDUs are used. This would imply dynamic length IPDUs should be possible.

  Without dynamic length support it is also not possible to support securing complete IDPUM Containers.
  One possible solution could be (if dynamic length should not be supported by SecOC) to add a configuration option per IPDUM Container to send always maximum length (padding with 0) to have the static length again.
  If dynamic length shall be supported by SecOC a further problem will be the CAN-FD padding.

  Agreed solution:

==
AUTOSAR 4.3.1
==

SRS SecOC

----

* Add new section next to 6.1.3.5 and new requirement [SRS_SecOC_xxxx1] Support of padding at lower layer modules and dynamic length Authentic I-PDUs.
* Description: The SecOC module shall be applicable for the use cases with padding at lower layer modules and with dynamic length Authentic I-PDUs.
* Rationale: At receiver side, received Secured I-PDU containing dynamic length Authentic I-PDU may also conatin padding bytes (added by lower layer modules of sender side, to fit to bus-specific L-PDU length constraints, e.g. CAN FD and FlexRay). In such case, receivers cannot identify number of bytes / byte position of the received payload.
* Use Case: dynamic length PDU on CAN FD and FlexRay
* Dependencies: [SRS_SecOC_00012]
* Supporting Material: -

Note: According to CM, RS Main should be referred. But upstream requirements in current (R4.3.0) SRS SecOC are from RS Features, and appropriate requirements are not available in RS Main.
(If we use RS Features, at least [RS_BRF_01568] [RS_BRF_01649] [RS_BRF_01712] [RS_BRF_01716] [RS_BRF_01752] [RS_BRF_02035] [RS_BRF_02036] [RS_BRF_02037] could be mapped to this requirements)

SWS SecOC

----

a1) Adapt Figure 4 in sec. 7.1.1.1

Change from
< (Figure of "Secured I-PDU = Authentic I-PDU | Freshness Value | Authenticator")
< Figure 4: Secured I-PDU contents
to
> (Figure of "Secured I-PDU = Secured I-PDU Header (optional) | Authentic I-PDU | Freshness Value (optional) | Authenticator")
> Figure 4: Secured I-PDU contents

==> to be done as RfC # 77807, not handled in this RfC.

a2) Add new requirements regarding the Secured I-PDU Header and related behavior

Add new requirement [SWS_SecOC_xxxx2] to define the Secured I-PDU Header
> The Secured I-PDU Header shall indicate the length of the Authentic I-PDU in bytes. The length of the Header shall be configurable by the parameter SecOCAuthPduHeaderLength.
> ()
> Note: the SecOC supports combined usage of authentication data in a separate message (secured PDU collection) and Secured I-PDU Header. Also the SecOC covers dynamic length Authentic I-PDU.

Add new requirement [SWS_SecOC_xxxx3] for behavior at transmission (construction) of Secured I-PDUs
> For a Tx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall add the Secured I-PDU Header to the Secured I-PDU with the length of the Authentic I-PDU within the Secured I-PDU, to handle dynamic Authentic I-PDU.
> ()
> Note: Primary purpose of this Header is to indicate the position of Freshness Value and Authenticator in Secured I-PDUs with dynamic length Authentic I-PDU.
> Also some buses which cannot select arbitrary length of L-PDU (e.g. CAN FD and FlexRay) require this Header, because the position of Freshness Value and Authenticator is not always at the end of the Secured I-PDU, as lower layer modules (e.g. CanIf and FrIf) may add bus-specific padding bytes after processing at SecOC (then the L-PDU containing the Secured I-PDU with padding will be: Secured I-PDU = Secured I-PDU Header | Authentic I-PDU | Freshness Value | Authenticator | Bus-specific padding).

Add new requirement [SWS_SecOC_xxxx4] for behavior at reception of Secured I-PDUs
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall assume Secured I-PDU Header shall be available in the Secured I-PDU, to handle dynamic Authentic I-PDU.
> ()

Add new requirement [SWS_SecOC_xxxx5] for behavior at reception of Se-

cured I-PDUs, the Header tells it's longer than the maximum length of the PDU
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0 and the length of Authentic I-PDU in the Header is longer than configured length (in case of dynamic length IPdus (containing a dynamical length signal), this value indicates the maximum data length) of the Authentic I-PDU, the SecOC module shall discard the I-PDU. In such case with SecOC_StartOfReception, BUFREQ_E_NOT_OK shall be returned (see [SWS_COMTYPE_00012]).
> ()
> Note: SecOC_RxIndication has no return value.

Add new requirement [SWS_SecOC_xxxx6] for behavior at reception of Secured I-PDUs, the Header tells it's shorter than received I-PDU length (ignoring the padding at the end)
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall process Secured I-PDU Header, Authentic I-PDU (with the length specified by the Header), Freshness Value and Authenticator of the Rx Secured I-PDU. The rest of bytes in the Secured I-PDU shall be discarded.
> ()

a3) Add a configuration parameter to SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu, SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxAuthenticPdu, SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPdu and SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu to enable/disable Secured I-PDU Header per I-PDU
* SWS Item: ECUC_SecOC_xxxx1
* Name: SecOCAuthPduHeaderLength
* Description:
* This parameter indicates the length (in bytes) of the Secured I-PDU Header in the Secured I-PDU. The length of zero means there's no header in the PDU.
* Multiplicity: 0..1
* Range: 0..4
* Default: 0

a4) (removed)

a8) Update layout definition (construction) for Secured I-PDUs

Change [SWS_SecOC_00037]
from
< [SWS_SecOC_00037]
< The SecOC module shall construct the Secured I-PDU by adding the Freshness Value and the Authenticator to the Authentic I-PDU.
< (SRS_SecOC_00006)
< Note: The Freshness Counter and the Authenticator included as part of the Secured I-PDU may be truncated per configuration specific to the identifier of the Secured I-PDU. The scheme for the Secured I-PDU looks as follows:
< SecuredPDU = AuthenticIPDU | FreshnessValue [SecOCFreshnessValueTxLength] | Authenticator [SecOCAuthInfoTxLength]
to
> [SWS_SecOC_00037]
> The SecOC module shall construct the Secured I-PDU by adding the Secured I-PDU Header (optional), the Freshness Value (optional) and the Authenticator to the Authentic I-PDU.
> The scheme for the Secured I-PDU (includes the order in which the contents are structured in the Secured I-PDU) shall be compliant with below:
> SecuredPDU = SecuredIPDUHeader (optional) | AuthenticIPDU | FreshnessValue [SecOCFreshnessValueTxLength] (optional) | Authenticator [SecOCAuthInfoTxLength]
> (SRS_SecOC_00006)
> Note: The Freshness Counter and the Authenticator included as part of the Secured I-PDU may be truncated per configuration specific to the identifier of the Secured I-PDU. Also Freshness Value may be a part of Authentic I-PDU (see [SWS_SecOC_00219]).
==> to be done as RfC # 77807, not handled in this RfC.

a9) Add new constraints and notes after [SWS_SecOC_00219]:
> [constr_xxxx1] All signals before SecOCAuthDataFreshnessStartPosition within the Secured I-PDU shall have static length.
> Note: SecOC can use a part of the Authentic I-PDU as freshness when SecOCUseAuthDataFreshness=true, only if the part of the Authentic I-PDU to be used as the freshness is always available at same position in the Authentic I-PDU.

> [constr_xxxx2] Any container I-PDU which contains multiple contained I-PDUs shall be set SecOCUseAuthDataFreshness=false.
> Note: For container PDUs, normally it cannot be ensured which PDU will be put in which position (depends on various timing and trigger conditions). Therefore, container I-PDUs with multiple contained I-PDUs cannot have FV within the Authentic I-PDU.

a10) Adapt Figure 5
==> to be done as RfC # 77807, not handled in this RfC.

a11) Adapt SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxAuthenticPdu and SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu

Change the description of SecOCRxAuthenticPdu (ECUC_SecOC_00061)
from
< This container specifies the Authetic Pdu that is received by the SecOC module from the PduR.
to
> This container specifies the PDU (that is received by the SecOC module from the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.

Change the description of SecOCTxAuthenticPdu ECUC_SecOC_00072
from
< This container specifies the Authetic Pdu that is transmitted by the SecOC module to the PduR after the Mac was generated.
to
> This container specifies the PDU (that is transmitted by the SecOC module to the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.

—————-
TPS System Template (SysT)
—————-

a5) Add new attribute to SecuredIPdu (Table 6.46) which enables SecOCAuthPduHeaderLength>0
* Attribute: useSecuredPduHeader
* Type: SecuredPduHeaderEnum
* Mul.: 0..1
* Kind: attr
* Desc: This attribute defines the size of the header which is inserted into the SecuredIPdu. If this attribute is set to anything but noHeader, the SecuredIPdu contains the Secured I-PDU Header to indicate the length of the AuthenticIPdu. The AuthenticIPdu contains the original payload, i.e. the secured data.

SecuredPduHeaderEnum
- noHeader
- securedPduHeader08Bit
- securedPduHeader16Bit
- securedPduHeader32Bit

Desc: Defines the header which will be inserted into the SecuredIPdu.

a6) Change the description of IPduPort.rxSecurityVerification in Table 6.3: IPduPort:
This attribute defines the bypassing of signature authentication or MAC verification in the receiving ECU.
If not defined or set to true the signature authentication or MAC verification shall be performed for the SecuredIPdu.
If set to false the signature authentication or MAC verification shall not be performed for the SecuredIPdu.

Removed [constr_3139].

TPS_SysT_xxxx2: Setting of useSecuredPduHeader attribute
The useSecuredPduHeader shall be set to a value other than noHeader if the length of the payload Pdu is dynamic and is transmitted over a network which may insert padding bytes depending on the length (e.g. CANFD, Flexray).

Add a note below TPS_SysT_xxxx2:

Please note that the dynamic-length Pdu can be an ISignalIPdu that contains a SystemSignal with dynamicLength set to true. In general it is not possible to run diagnostics on fixed-length Pdus. Therefore, there is a probability that at least a subset of DcmIPdus and UserDefinedIPdus can have dynamic length.

a7) Add upstream mapping between useSecuredPduHeader (SysT) and SecOCAuthPduHeaderLength (EcuC) in C.1.4 SecOc Mapping

b4) Add upstream mapping between rxSecurityVerification (SysT) and SecOCSecuredRxPduVerification (EcuC) in C.1.4 SecOc Mapping
Mapping rule: SecOCSecuredRxPduVerification is True if rxSecurityVerification is not defined, otherwise SecOCSecuredRxPduVerification = rxSecurityVerification

——————-

SRS SecOC

——————-

* Add new section next to 6.1.3.5 (or 6.2.1.1) and new requirement

[SRS_SecOC_xxxx2] Support of capability to extract Authentic I-PDU without Authentication
* Description: The SecOC module shall be capable to extract Authentic I-PDU from Secured I-PDU, without Authentication.
* Rationale: SecOC can be used as an extractor of Authentic I-PDU from Secured I-PDU, to enable low latency GW behavior when a part of downstream communication clusters doesn't require authentication of PDUs.
* Use Case: Gateway
* Dependencies: [SRS_SecOC_00025]
* Supporting Material: -

Note: According to CM, RS Main should be referred. But upstream requirments in current (R4.3.0) SRS SecOC are from RS Features, and appropriate requirements are not available in RS Main.
(If we use RS Features, at least [RS_BRF_02035] [RS_BRF_02036] [RS_BRF_02037] could be mapped to this requirements)

——————-
SWS SecOC
——————-

b1) Remove [constr_3139] (not [constr_3193] – sorry, constr_3193 is typo in my comment # 10)

b2) Add new requirements regarding skipped authentication behavior at SecOC (just remove FV/MAC from Secured I-PDU)

* Add new section "Extracting Authentic I-PDU without Authentication at SecOC" or "Skipping Authentication for Secured I-PDUs at SecOC"
* Add new requirement [SWS_SecOC_xxxx7] for behavior of SecOC at reception of Secured I-PDUs without Authentication
> For a Rx Secured I-PDU with SecOCSecuredRxPduVerification=false, the SecOC module shall extract the Authentic I-PDU using the length specified by the Secured I-PDU Header without Authentication.
> ()

b3) Add a configuration parameter to SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu and SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection to control authentication behavior at SecOC

* SWS Item: ECUC_SecOC_xxxx3
* Name: SecOCSecuredRxPduVerification
* Description: This parameter defines whether the signature authentication or MAC verification shall be performed on this Secured I-PDU. If set to false, the SecOC module extracts the Authentic I-PDU from the Secured I-PDU without verification.
* Multiplicity: 1
* Type: EcucBooleanParamDef
* Default value: false
* Post-Build Variant Value: true
* Value Configuration Class:
* Pre-compile time: X All Variants
* Scope / Dependency: scope: local
–Last change on issue 77336 comment 69–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.37 Specification Item SWS_SecOC_00264

Trace References:

SRS_BSW_00385

Content:

For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall process Secured I-PDU Header, Authentic I-PDU (with the length specified by the Header), Freshness Value and Authenticator of the Rx Secured I-PDU. The rest of bytes in the Secured I-PDU shall be discarded.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77336: [SECOC] Dynamic length PDUs (Container) not possible / clear

  Problem description:

  The SecOC SWS does not make any assumptions or restrictions about dynamic length PDUs. But with parameter SecOCAuthDataFreshnessStartPosition it could be impossible to really use dynamic length in SecOC.

  Additionally in SystemTemplate TPS the constraint constr_3139 talks only about some restrictions when dynamic length IPDUs are used. This would imply dynamic length IPDUs should be possible.

Without dynamic length support it is also not possible to support securing complete IDPUM Containers.
One possible solution could be (if dynamic length should not be supported by SecOC) to add a configuration option per IPDUM Container to send always maximum length (padding with 0) to have the static length again.
If dynamic length shall be supported by SecOC a further problem will be the CAN-FD padding.

Agreed solution:

==
AUTOSAR 4.3.1
==

———————-
SRS SecOC
———————-

* Add new section next to 6.1.3.5 and new requirement [SRS_SecOC_xxxx1] Support of padding at lower layer modules and dynamic length Authentic I-PDUs.
* Description: The SecOC module shall be applicable for the use cases with padding at lower layer modules and with dynamic length Authentic I-PDUs.
* Rationale: At receiver side, received Secured I-PDU containing dynamic length Authentic I-PDU may also conatin padding bytes (added by lower layer modules of sender side, to fit to bus-specific L-PDU length constraints, e.g. CAN FD and FlexRay). In such case, receivers cannot identify number of bytes / byte position of the received payload.
* Use Case: dynamic length PDU on CAN FD and FlexRay
* Dependencies: [SRS_SecOC_00012]
* Supporting Material: -

Note: According to CM, RS Main should be referred. But upstream requirements in current (R4.3.0) SRS SecOC are from RS Features, and appropriate requirements are not available in RS Main.
(If we use RS Features, at least [RS_BRF_01568] [RS_BRF_01649] [RS_BRF_01712] [RS_BRF_01716] [RS_BRF_01752] [RS_BRF_02035] [RS_BRF_02036] [RS_BRF_02037] could be mapped to this requirements)

———————-
SWS SecOC
———————-

a1) Adapt Figure 4 in sec. 7.1.1.1

Change from
< (Figure of "Secured I-PDU = Authentic I-PDU | Freshness Value | Authenticator")
< Figure 4: Secured I-PDU contents
to
> (Figure of "Secured I-PDU = Secured I-PDU Header (optional) | Authentic I-PDU | Freshness Value (optional) | Authenticator")
> Figure 4: Secured I-PDU contents
==> to be done as RfC # 77807, not handled in this RfC.

a2) Add new requirements regarding the Secured I-PDU Header and related behavior

Add new requirement [SWS_SecOC_xxxx2] to define the Secured I-PDU Header
> The Secured I-PDU Header shall indicate the length of the Authentic I-PDU in bytes. The length of the Header shall be configurable by the parameter SecOCAuthPduHeaderLength.
> ()
> Note: the SecOC supports combined usage of authentication data in a separate message (secured PDU collection) and Secured I-PDU Header. Also the SecOC covers dynamic length Authentic I-PDU.

Add new requirement [SWS_SecOC_xxxx3] for behavior at transmission (construction) of Secured I-PDUs
> For a Tx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall add the Secured I-PDU Header to the Secured I-PDU with the length of the Authentic I-PDU within the Secured I-PDU, to handle dynamic Authentic I-PDU.
> ()
> Note: Primary purpose of this Header is to indicate the position of Freshness Value and Authenticator in Secured I-PDUs with dynamic length Authentic I-PDU.
> Also some buses which cannot select arbitrary length of L-PDU (e.g. CAN FD and FlexRay) require this Header, because the position of Freshness Value and Authenticator is not always at the end of the Secured I-PDU, as lower layer modules (e.g. CanIf and FrIf) may add bus-specific padding bytes after processing at SecOC (then the L-PDU containing the Secured I-PDU with padding will be: Secured I-PDU = Secured I-PDU Header | Authentic I-PDU | Freshness Value | Authenticator | Bus-specific padding).

Add new requirement [SWS_SecOC_xxxx4] for behavior at reception of Secured I-PDUs
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall assume Secured I-PDU Header shall be available in the Secured I-PDU, to handle dynamic Authentic I-PDU.
> ()

Add new requirement [SWS_SecOC_xxxx5] for behavior at reception of Secured I-PDUs, the Header tells it's longer than the maximum length of the PDU
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0 and the length of Authentic I-PDU in the Header is longer than configured length (in case of dynamic length IPdus (containing a dynamical length signal), this value indicates the maximum data length) of the Authentic I-PDU, the SecOC module shall discard the I-PDU. In such case with SecOC_StartOfReception, BUFREQ_E_NOT_OK shall be returned (see [SWS_COMTYPE_00012]).
> ()
> Note: SecOC_RxIndication has no return value.

Add new requirement [SWS_SecOC_xxxx6] for behavior at reception of Secured I-PDUs, the Header tells it's shorter than received I-PDU length (ignoring the padding at the end)
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall process Secured I-PDU Header, Authentic I-PDU (with the length specified by the Header), Freshness Value and Authenticator of the Rx Secured I-PDU. The rest of bytes in the Secured I-PDU shall be discarded.
> ()

a3) Add a configuration parameter to SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu, SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxAuthenticPdu, SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPdu and SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu to enable/disable Secured I-PDU Header per I-PDU
* SWS Item: ECUC_SecOC_xxxx1
* Name: SecOCAuthPduHeaderLength
* Description:
* This parameter indicates the length (in bytes) of the Secured I-PDU Header in the

Secured I-PDU. The length of zero means there's no header in the PDU.
* Multiplicity: 0..1
* Range: 0..4
* Default: 0

a4) (removed)

a8) Update layout definition (construction) for Secured I-PDUs

Change [SWS_SecOC_00037]
from
< [SWS_SecOC_00037]
< The SecOC module shall construct the Secured I-PDU by adding the Freshness Value and the Authenticator to the Authentic I-PDU.
< (SRS_SecOC_00006)
< Note: The Freshness Counter and the Authenticator included as part of the Secured I-PDU may be truncated per configuration specific to the identifier of the Secured I-PDU. The scheme for the Secured I-PDU looks as follows:
< SecuredPDU = AuthenticIPDU | FreshnessValue [SecOCFreshnessValueTxLength] | Authenticator [SecOCAuthInfoTxLength]
to
> [SWS_SecOC_00037]
> The SecOC module shall construct the Secured I-PDU by adding the Secured I-PDU Header (optional), the Freshness Value (optional) and the Authenticator to the Authentic I-PDU.
> The scheme for the Secured I-PDU (includes the order in which the contents are structured in the Secured I-PDU) shall be compliant with below:
> SecuredPDU = SecuredIPDUHeader (optional) | AuthenticIPDU | FreshnessValue [SecOCFreshnessValueTxLength] (optional) | Authenticator [SecOCAuthInfoTxLength]
> (SRS_SecOC_00006)
> Note: The Freshness Counter and the Authenticator included as part of the Secured I-PDU may be truncated per configuration specific to the identifier of the Secured I-PDU. Also Freshness Value may be a part of Authentic I-PDU (see [SWS_SecOC_00219]).
==> to be done as RfC # 77807, not handled in this RfC.

a9) Add new constraints and notes after [SWS_SecOC_00219]:
> [constr_xxxx1] All signals before SecOCAuthDataFreshnessStartPosition within the Secured I-PDU shall have static length.

> Note: SecOC can use a part of the Authentic I-PDU as freshness when SecOCUseAuthDataFreshness=true, only if the part of the Authentic I-PDU to be used as the freshness is always available at same position in the Authentic I-PDU.

> [constr_xxxx2] Any container I-PDU which contains multiple contained I-PDUs shall be set SecOCUseAuthDataFreshness=false.
> Note: For container PDUs, normally it cannot be ensured which PDU will be put in which position (depends on various timing and trigger conditions). Therefore, container I-PDUs with multiple contained I-PDUs cannot have FV within the Authentic I-PDU.

a10) Adapt Figure 5
==> to be done as RfC # 77807, not handled in this RfC.

a11) Adapt SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxAuthenticPdu and SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu

Change the description of SecOCRxAuthenticPdu (ECUC_SecOC_00061)
from
< This container specifies the Authetic Pdu that is received by the SecOC module from the PduR.
to
> This container specifies the PDU (that is received by the SecOC module from the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.

Change the description of SecOCTxAuthenticPdu ECUC_SecOC_00072
from
< This container specifies the Authetic Pdu that is transmitted by the SecOC module to the PduR after the Mac was generated.
to
> This container specifies the PDU (that is transmitted by the SecOC module to the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.

————————-
TPS System Template (SysT)
————————-

a5) Add new attribute to SecuredIPdu (Table 6.46) which enables SecOCAu-

thPduHeaderLength>0
* Attribute: useSecuredPduHeader
* Type: SecuredPduHeaderEnum
* Mul.: 0..1
* Kind: attr
* Desc: This attribute defines the size of the header which is inserted into the SecuredIPdu. If this attribute is set to anything but noHeader, the SecuredIPdu contains the Secured I-PDU Header to indicate the length of the AuthenticIPdu. The AuthenticIPdu contains the original payload, i.e. the secured data.

SecuredPduHeaderEnum
- noHeader
- securedPduHeader08Bit
- securedPduHeader16Bit
- securedPduHeader32Bit
Desc: Defines the header which will be inserted into the SecuredIPdu.

a6) Change the description of IPduPort.rxSecurityVerification in Table 6.3: IPduPort:
This attribute defines the bypassing of signature authentication or MAC verification in the receiving ECU.
If not defined or set to true the signature authentication or MAC verification shall be performed for the SecuredIPdu.
If set to false the signature authentication or MAC verification shall not be performed for the SecuredIPdu.

Removed [constr_3139].

TPS_SysT_xxxx2: Setting of useSecuredPduHeader attribute
The useSecuredPduHeader shall be set to a value other than noHeader if the length of the payload Pdu is dynamic and is transmitted over a network which may insert padding bytes depending on the length (e.g. CANFD, Flexray).

Add a note below TPS_SysT_xxxx2:

Please note that the dynamic-length Pdu can be an ISignalIPdu that contains a SystemSignal with dynamicLength set to true. In general it is not possible to run diagnostics on fixed-length Pdus. Therefore, there is a probability that at least a subset of DcmIPdus and UserDefinedIPdus can have dynamic length.

a7) Add upstream mapping between useSecuredPduHeader (SysT) and SecOCAuthPduHeaderLength (EcuC) in C.1.4 SecOc Mapping

b4) Add upstream mapping between rxSecurityVerification (SysT) and SecOCSecuredRxPduVerification (EcuC) in C.1.4 SecOc Mapping
Mapping rule: SecOCSecuredRxPduVerification is True if rxSecurityVerification is not defined, otherwise SecOCSecuredRxPduVerification = rxSecurityVerification

——————-

SRS SecOC

——————-

* Add new section next to 6.1.3.5 (or 6.2.1.1) and new requirement [SRS_SecOC_xxxx2] Support of capability to extract Authentic I-PDU without Authentication
* Description: The SecOC module shall be capable to extract Authentic I-PDU from Secured I-PDU, without Authentication.
* Rationale: SecOC can be used as an extractor of Authentic I-PDU from Secured I-PDU, to enable low latency GW behavior when a part of downstream communication clusters doesn't require authentication of PDUs.
* Use Case: Gateway
* Dependencies: [SRS_SecOC_00025]
* Supporting Material: -

Note: According to CM, RS Main should be referred. But upstream requirments in current (R4.3.0) SRS SecOC are from RS Features, and appropriate requirements are not available in RS Main.
(If we use RS Features, at least [RS_BRF_02035] [RS_BRF_02036] [RS_BRF_02037] could be mapped to this requirements)

——————-

SWS SecOC

——————-

b1) Remove [constr_3139] (not [constr_3193] – sorry, constr_3193 is typo in my comment # 10)

b2) Add new requirements regarding skipped authentication behavior at SecOC (just remove FV/MAC from Secured I-PDU)

* Add new section "Extracting Authentic I-PDU without Authentication at SecOC" or "Skipping Authentication for Secured I-PDUs at SecOC"
* Add new requirement [SWS_SecOC_xxxx7] for behavior of SecOC at reception of Secured I-PDUs without Authentication

> For a Rx Secured I-PDU with SecOCSecuredRxPduVerification=false, the SecOC module shall extract the Authentic I-PDU using the length specified by the Secured I-PDU Header without Authentication.
> ()

b3) Add a configuration parameter to SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu and SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection to control authentication behavior at SecOC

* SWS Item: ECUC_SecOC_xxxx3
* Name: SecOCSecuredRxPduVerification
* Description: This parameter defines whether the signature authentication or MAC verification shall be performed on this Secured I-PDU. If set to false, the SecOC module extracts the Authentic I-PDU from the Secured I-PDU without verification.
* Multiplicity: 1
* Type: EcucBooleanParamDef
* Default value: false
* Post-Build Variant Value: true
* Value Configuration Class:
* Pre-compile time: X All Variants
* Scope / Dependency: scope: local
–Last change on issue 77336 comment 69–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.38 Specification Item SWS_SecOC_00265

Trace References:

SRS_BSW_00385

Content:

For a Rx Secured I-PDU with SecOCSecuredRxPduVerification=false, the SecOC module shall extract the Authentic I-PDU using the length specified by the Secured I-PDU Header without Authentication.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77336: [SECOC] Dynamic length PDUs (Container) not possible / clear

Problem description:

The SecOC SWS does not make any assumptions or restrictions about dynamic length PDUs. But with parameter SecOCAuthDataFreshnessStartPosition it could be impossible to really use dynamic length in SecOC.

Additionally in SystemTemplate TPS the constraint constr_3139 talks only about some restrictions when dynamic length IPDUs are used. This would imply dynamic length IPDUs should be possible.

Without dynamic length support it is also not possible to support securing complete IDPUM Containers.
One possible solution could be (if dynamic length should not be supported by SecOC) to add a configuration option per IPDUM Container to send always maximum length (padding with 0) to have the static length again.
If dynamic length shall be supported by SecOC a further problem will be the CAN-FD padding.

Agreed solution:

===
AUTOSAR 4.3.1
===

———————-
SRS SecOC
———————-

* Add new section next to 6.1.3.5 and new requirement [SRS_SecOC_xxxx1] Support of padding at lower layer modules and dynamic length Authentic I-PDUs.
* Description: The SecOC module shall be applicable for the use cases with padding at lower layer modules and with dynamic length Authentic I-PDUs.
* Rationale: At receiver side, received Secured I-PDU containing dynamic length Authentic I-PDU may also conatin padding bytes (added by lower layer modules of sender side, to fit to bus-specific L-PDU length constraints, e.g. CAN FD and FlexRay). In such case, receivers cannot identify number of bytes / byte position of the received payload.
* Use Case: dynamic length PDU on CAN FD and FlexRay
* Dependencies: [SRS_SecOC_00012]
* Supporting Material: -

Note: According to CM, RS Main should be referred. But upstream requirements in current (R4.3.0) SRS SecOC are from RS Features, and appropriate

requirements are not available in RS Main.
(If we use RS Features, at least [RS_BRF_01568] [RS_BRF_01649] [RS_BRF_01712] [RS_BRF_01716] [RS_BRF_01752] [RS_BRF_02035] [RS_BRF_02036] [RS_BRF_02037] could be mapped to this requirements)

—————-
SWS SecOC
—————-

a1) Adapt Figure 4 in sec. 7.1.1.1

Change from
< (Figure of "Secured I-PDU = Authentic I-PDU | Freshness Value | Authenticator")
< Figure 4: Secured I-PDU contents
to
> (Figure of "Secured I-PDU = Secured I-PDU Header (optional) | Authentic I-PDU | Freshness Value (optional) | Authenticator")
> Figure 4: Secured I-PDU contents
==> to be done as RfC # 77807, not handled in this RfC.

a2) Add new requirements regarding the Secured I-PDU Header and related behavior

Add new requirement [SWS_SecOC_xxxx2] to define the Secured I-PDU Header
> The Secured I-PDU Header shall indicate the length of the Authentic I-PDU in bytes. The length of the Header shall be configurable by the parameter SecOCAuthPduHeaderLength.
> ()
> Note: the SecOC supports combined usage of authentication data in a separate message (secured PDU collection) and Secured I-PDU Header. Also the SecOC covers dynamic length Authentic I-PDU.

Add new requirement [SWS_SecOC_xxxx3] for behavior at transmission (construction) of Secured I-PDUs
> For a Tx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall add the Secured I-PDU Header to the Secured I-PDU with the length of the Authentic I-PDU within the Secured I-PDU, to handle dynamic Authentic I-PDU.
> ()
> Note: Primary purpose of this Header is to indicate the position of Freshness

Value and Authenticator in Secured I-PDUs with dynamic length Authentic I-PDU.
> Also some buses which cannot select arbitrary length of L-PDU (e.g. CAN FD and FlexRay) require this Header, because the position of Freshness Value and Authenticator is not always at the end of the Secured I-PDU, as lower layer modules (e.g. CanIf and FrIf) may add bus-specific padding bytes after processing at SecOC (then the L-PDU containing the Secured I-PDU with padding will be: Secured I-PDU = Secured I-PDU Header | Authentic I-PDU | Freshness Value | Authenticator | Bus-specific padding).

Add new requirement [SWS_SecOC_xxxx4] for behavior at reception of Secured I-PDUs
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall assume Secured I-PDU Header shall be available in the Secured I-PDU, to handle dynamic Authentic I-PDU.
> ()

Add new requirement [SWS_SecOC_xxxx5] for behavior at reception of Secured I-PDUs, the Header tells it's longer than the maximum length of the PDU
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0 and the length of Authentic I-PDU in the Header is longer than configured length (in case of dynamic length IPdus (containing a dynamical length signal), this value indicates the maximum data length) of the Authentic I-PDU, the SecOC module shall discard the I-PDU. In such case with SecOC_StartOfReception, BUFREQ_E_NOT_OK shall be returned (see [SWS_COMTYPE_00012]).
> ()
> Note: SecOC_RxIndication has no return value.

Add new requirement [SWS_SecOC_xxxx6] for behavior at reception of Secured I-PDUs, the Header tells it's shorter than received I-PDU length (ignoring the padding at the end)
> For a Rx Secured I-PDU with SecOCAuthPduHeaderLength > 0, the SecOC module shall process Secured I-PDU Header, Authentic I-PDU (with the length specified by the Header), Freshness Value and Authenticator of the Rx Secured I-PDU. The rest of bytes in the Secured I-PDU shall be discarded.
> ()

a3) Add a configuration parameter to SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu, SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSe-

curedPduCollection/SecOCRxAuthenticPdu, SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPdu and SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu to enable/disable Secured I-PDU Header per I-PDU
* SWS Item: ECUC_SecOC_xxxx1
* Name: SecOCAuthPduHeaderLength
* Description:
* This parameter indicates the length (in bytes) of the Secured I-PDU Header in the Secured I-PDU. The length of zero means there's no header in the PDU.
* Multiplicity: 0..1
* Range: 0..4
* Default: 0

a4) (removed)

a8) Update layout definition (construction) for Secured I-PDUs

Change [SWS_SecOC_00037]
from
< [SWS_SecOC_00037]
< The SecOC module shall construct the Secured I-PDU by adding the Freshness Value and the Authenticator to the Authentic I-PDU.
< (SRS_SecOC_00006)
< Note: The Freshness Counter and the Authenticator included as part of the Secured I-PDU may be truncated per configuration specific to the identifier of the Secured I-PDU. The scheme for the Secured I-PDU looks as follows:
< SecuredPDU = AuthenticIPDU | FreshnessValue [SecOCFreshnessValueTxLength] | Authenticator [SecOCAuthInfoTxLength]
to
> [SWS_SecOC_00037]
> The SecOC module shall construct the Secured I-PDU by adding the Secured I-PDU Header (optional), the Freshness Value (optional) and the Authenticator to the Authentic I-PDU.
> The scheme for the Secured I-PDU (includes the order in which the contents are structured in the Secured I-PDU) shall be compliant with below:
> SecuredPDU = SecuredIPDUHeader (optional) | AuthenticIPDU | FreshnessValue [SecOCFreshnessValueTxLength] (optional) | Authenticator [SecOCAuthInfoTxLength]
> (SRS_SecOC_00006)
> Note: The Freshness Counter and the Authenticator included as part of the

Secured I-PDU may be truncated per configuration specific to the identifier of the Secured I-PDU. Also Freshness Value may be a part of Authentic I-PDU (see [SWS_SecOC_00219]).
==> to be done as RfC # 77807, not handled in this RfC.

a9) Add new constraints and notes after [SWS_SecOC_00219]:
> [constr_xxxx1] All signals before SecOCAuthDataFreshnessStartPosition within the Secured I-PDU shall have static length.
> Note: SecOC can use a part of the Authentic I-PDU as freshness when SecOCUseAuthDataFreshness=true, only if the part of the Authentic I-PDU to be used as the freshness is always available at same position in the Authentic I-PDU.

> [constr_xxxx2] Any container I-PDU which contains multiple contained I-PDUs shall be set SecOCUseAuthDataFreshness=false.
> Note: For container PDUs, normally it cannot be ensured which PDU will be put in which position (depends on various timing and trigger conditions). Therefore, container I-PDUs with multiple contained I-PDUs cannot have FV within the Authentic I-PDU.

a10) Adapt Figure 5
==> to be done as RfC # 77807, not handled in this RfC.

a11) Adapt SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxAuthenticPdu and SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu

Change the description of SecOCRxAuthenticPdu (ECUC_SecOC_00061)
from
< This container specifies the Authetic Pdu that is received by the SecOC module from the PduR.
to
> This container specifies the PDU (that is received by the SecOC module from the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.

Change the description of SecOCTxAuthenticPdu ECUC_SecOC_00072
from
< This container specifies the Authetic Pdu that is transmitted by the SecOC module to the PduR after the Mac was generated.
to

> This container specifies the PDU (that is transmitted by the SecOC module to the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.

——————-
TPS System Template (SysT)
——————-

a5) Add new attribute to SecuredIPdu (Table 6.46) which enables SecOCAuthPduHeaderLength>0
* Attribute: useSecuredPduHeader
* Type: SecuredPduHeaderEnum
* Mul.: 0..1
* Kind: attr
* Desc: This attribute defines the size of the header which is inserted into the SecuredIPdu. If this attribute is set to anything but noHeader, the SecuredIPdu contains the Secured I-PDU Header to indicate the length of the AuthenticIPdu. The AuthenticIPdu contains the original payload, i.e. the secured data.

SecuredPduHeaderEnum
- noHeader
- securedPduHeader08Bit
- securedPduHeader16Bit
- securedPduHeader32Bit
Desc: Defines the header which will be inserted into the SecuredIPdu.

a6) Change the description of IPduPort.rxSecurityVerification in Table 6.3: IPduPort:
This attribute defines the bypassing of signature authentication or MAC verification in the receiving ECU.
If not defined or set to true the signature authentication or MAC verification shall be performed for the SecuredIPdu.
If set to false the signature authentication or MAC verification shall not be performed for the SecuredIPdu.

Removed [constr_3139].

TPS_SysT_xxxx2: Setting of useSecuredPduHeader attribute
The useSecuredPduHeader shall be set to a value other than noHeader if the length of the payload Pdu is dynamic and is transmitted over a network which may insert padding bytes depending on the length (e.g. CANFD, Flexray).

Add a note below TPS_SysT_xxxx2:

Please note that the dynamic-length Pdu can be an ISignalIPdu that contains a SystemSignal with dynamicLength set to true. In general it is not possible to run diagnostics on fixed-length Pdus. Therefore, there is a probability that at least a subset of DcmIPdus and UserDefinedIPdus can have dynamic length.

a7) Add upstream mapping between useSecuredPduHeader (SysT) and SecOCAuthPduHeaderLength (EcuC) in C.1.4 SecOc Mapping

b4) Add upstream mapping between rxSecurityVerification (SysT) and SecOCSecuredRxPduVerification (EcuC) in C.1.4 SecOc Mapping
Mapping rule: SecOCSecuredRxPduVerification is True if rxSecurityVerification is not defined, otherwise SecOCSecuredRxPduVerification = rxSecurityVerification

———————-
SRS SecOC
———————-

* Add new section next to 6.1.3.5 (or 6.2.1.1) and new requirement [SRS_SecOC_xxxx2] Support of capability to extract Authentic I-PDU without Authentication
* Description: The SecOC module shall be capable to extract Authentic I-PDU from Secured I-PDU, without Authentication.
* Rationale: SecOC can be used as an extractor of Authentic I-PDU from Secured I-PDU, to enable low latency GW behavior when a part of downstream communication clusters doesn’t require authentication of PDUs.
* Use Case: Gateway
* Dependencies: [SRS_SecOC_00025]
* Supporting Material: -

Note: According to CM, RS Main should be referred. But upstream requirments in current (R4.3.0) SRS SecOC are from RS Features, and appropriate requirements are not available in RS Main.
(If we use RS Features, at least [RS_BRF_02035] [RS_BRF_02036] [RS_BRF_02037] could be mapped to this requirements)

———————-
SWS SecOC
———————-

b1) Remove [constr_3139] (not [constr_3193] – sorry, constr_3193 is typo in my comment # 10) b2) Add new requirements regarding skipped authentication behavior at SecOC (just remove FV/MAC from Secured I-PDU)

* Add new section "Extracting Authentic I-PDU without Authentication at SecOC" or "Skipping Authentication for Secured I-PDUs at SecOC"
* Add new requirement [SWS_SecOC_xxxx7] for behavior of SecOC at reception of Secured I-PDUs without Authentication
> For a Rx Secured I-PDU with SecOCSecuredRxPduVerification=false, the SecOC module shall extract the Authentic I-PDU using the length specified by the Secured I-PDU Header without Authentication.
> ()

b3) Add a configuration parameter to SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu and SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection to control authentication behavior at SecOC

* SWS Item: ECUC_SecOC_xxxx3
* Name: SecOCSecuredRxPduVerification
* Description: This parameter defines whether the signature authentication or MAC verification shall be performed on this Secured I-PDU. If set to false, the SecOC module extracts the Authentic I-PDU from the Secured I-PDU without verification.
* Multiplicity: 1
* Type: EcucBooleanParamDef
* Default value: false
* Post-Build Variant Value: true
* Value Configuration Class:
* Pre-compile time: X All Variants
* Scope / Dependency: scope: local
–Last change on issue 77336 comment 69–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.39 Specification Item SWS_SecOC_91002

Trace References:

SRS_SECOC_00003, SRS_SECOC_00021, SRS_SECOC_00022

Content:

Name	FreshnessManagementFreshnessManagement	
Comment	Freshness Management for SecOC	
IsService	true	
Variation	–	
Possible Errors	0	E_OK
	1	E_NOT_OK
	2	E_BUSY

Operations:

GetRxFreshnessFreshnessManagement.GetRxFreshness	
Comments	This interface is used by the SecOC to obtain the current freshness value. This operation provides also a part of the Authentic-PDU data if configured.
Variation	({ecuc(SecOC/SecOCRxPduProcessing/SecOCUseAuthDataFreshness)} == FALSE)

GetRxFreshnessFreshnessManagement.GetRxFreshness			
Parameters	freshnessValueIdFreshnessManagement.GetRxFreshness.freshnessValueId	Comment	Identifier of the freshness
		Type	uint16
		Variation	–
		Direction	IN
	truncatedFreshnessValueFreshnessManagement.GetRxFreshness.truncatedFreshnessValue	Comment	The truncated freshness value from the received Secured-IPDU
		Type	SecOC_FreshnessArrayType
		Variation	–
		Direction	IN
	truncatedFreshnessValueLengthFreshnessManagement.GetRxFreshness.truncatedFreshnessValueLength	Comment	Length in bits of the truncated freshness value
		Type	uint32
		Variation	–
		Direction	IN
	authVerifyAttemptsFreshnessManagement.GetRxFreshness.authVerifyAttempts	Comment	The number of authentication verify attempts for the current PDU
		Type	uint16
		Variation	–
		Direction	IN
	freshnessValueFreshnessManagement.GetRxFreshness.freshnessValue	Comment	The freshness value for this PDU
		Type	SecOC_FreshnessArrayType
		Variation	–
		Direction	OUT
	freshnessValueLengthFreshnessManagement.GetRxFreshness.freshnessValueLength	Comment	The freshness value length in bits.
		Type	uint32
		Variation	–
		Direction	INOUT
Possible Errors	E_OK	Operation successful	
	E_NOT_OK		
	E_BUSY	Operation temporary failed, a freshness cannot be provided at the moment.	

GetRxFreshnessAuthDataFreshnessManagement.GetRxFreshnessAuthData	
Comments	This interface is used by the SecOC to obtain the current freshness value. This operation provides also a part of the Authentic-PDU data if configured.
Variation	({ecuc(SecOC/SecOCRxPduProcessing/SecOCUseAuthDataFreshness)} == TRUE)

GetRxFreshnessAuthDataFreshnessManagement.GetRxFreshnessAuthData			
Parameters	freshnessValueIdFreshnessManagement.GetRxFreshnessAuthData.freshnessValueId	Comment	Identifier of the freshness
		Type	uint16
		Variation	–
		Direction	IN
	truncatedFreshnessValueFreshnessManagement.GetRxFreshnessAuthData.truncatedFreshnessValue	Comment	The truncated freshness value from the received Secured-IPDU
		Type	SecOC_FreshnessArrayType
		Variation	–
		Direction	IN
	truncatedFreshnessValueLengthFreshnessManagement.GetRxFreshnessAuthData.truncatedFreshnessValueLength	Comment	Length in bits of the truncated freshness value
		Type	uint32
		Variation	–
		Direction	IN
	authenticDataFreshnessValueFreshnessManagement.GetRxFreshnessAuthData.authenticDataFreshnessValue	Comment	The selected part of the authentic data.
		Type	SecOC_FreshnessArrayType
		Variation	–
		Direction	IN
	authenticDataFreshnessValueLengthFreshnessManagement.GetRxFreshnessAuthData.authenticDataFreshnessValueLength	Comment	The length in bits of the authentic data part.
		Type	uint16
		Variation	–
		Direction	IN
	authVerifyAttemptsFreshnessManagement.GetRxFreshnessAuthData.authVerifyAttempts	Comment	The number of authentication verify attempts for this PDU
		Type	uint16
		Variation	–
		Direction	IN
	freshnessValueFreshnessManagement.GetRxFreshnessAuthData.freshnessValue	Comment	The freshness value for this PDU
		Type	SecOC_FreshnessArrayType
		Variation	–
		Direction	OUT
	freshnessValueLengthFreshnessManagement.GetRxFreshnessAuthData.freshnessValueLength	Comment	The freshness value length in bits.
		Type	uint32
		Variation	–
		Direction	INOUT
Possible Errors	E_OK	Operation successful	
	E_NOT_OK		
	E_BUSY	Operation temporary failed, a freshness cannot be provided at the moment.	

GetTxFreshnessFreshnessManagement.GetTxFreshness			
Comments	This operation is used by the SecOC to obtain the freshness that corresponds to the freshness ValueIdReturns the freshness value from the Most Significant Bits in the first byte in the array (SecOCFreshnessValue), in big endian format.		
Variation	({ecuc(SecOC/SecOCTxPduProcessing/SecOCProvideTxTruncatedFreshnessValue)} == FALSE)		
Parameters	freshnessValueIdFreshnessManagement.GetTxFreshness.freshnessValueId	Comment	Identifier of the freshness
		Type	uint16
		Variation	–
		Direction	IN
	freshnessValueFreshnessManagement.GetTxFreshness.freshnessValue	Comment	Freshness value
		Type	SecOC_FreshnessArrayType
		Variation	–
		Direction	OUT
	freshnessValueLengthFreshnessManagement.GetTxFreshness.freshnessValueLength	Comment	Length in bits of the freshness value
		Type	uint32
		Variation	–
		Direction	INOUT
Possible Errors	E_OK	Operation successful	
	E_NOT_OK		
	E_BUSY	Operation temporary failed, a freshness cannot be provided at the moment.	

GetTxFreshnessTruncDataFreshnessManagement.GetTxFreshnessTruncData	
Comments	This operation is used by the SecOC to obtain the freshness that corresponds to the freshness ValueId. The operation provides the freshness and also the truncated freshness that shall be placed into the Secured-IPDU.
Variation	({ecuc(SecOC/SecOCTxPduProcessing/SecOCProvideTxTruncatedFreshnessValue)} == TRUE)

GetTxFreshnessTruncDataFreshnessManagement.GetTxFreshnessTruncData			
Parameters	freshnessValueIdFreshnessManagement.GetTxFreshnessTruncData.freshnessValueId	Comment	Identifier of the freshness
		Type	uint16
		Variation	–
		Direction	IN
	freshnessValueFreshnessManagement.GetTxFreshnessTruncData.freshnessValue	Comment	Freshness value
		Type	SecOC_FreshnessArrayType
		Variation	–
		Direction	OUT
	freshnessValueLengthFreshnessManagement.GetTxFreshnessTruncData.freshnessValueLength	Comment	Length in bits of the freshness value
		Type	uint32
		Variation	–
		Direction	INOUT
	truncatedFreshnessValueFreshnessManagement.GetTxFreshnessTruncData.truncatedFreshnessValue	Comment	The truncated freshness value that has to be placed into the Secured-IPDU
		Type	SecOC_FreshnessArrayType
		Variation	–
		Direction	OUT
	truncatedFreshnessValueLengthFreshnessManagement.GetTxFreshnessTruncData.truncatedFreshnessValueLength	Comment	The length in bits for the truncated freshness.
		Type	uint32
		Variation	–
		Direction	INOUT
Possible Errors	E_OK	Operation successful	
	E_NOT_OK		
	E_BUSY	Operation temporary failed, a freshness cannot be provided at the moment.	

SPduTxConfirmationFreshnessManagement.SPduTxConfirmation			
Comments	This operation is used by the SecOC to indicate that the Secured I-PDU has been initiated for transmission.		
Variation	–		
Parameters	freshnessValueIdFreshnessManagement.SPduTxConfirmation.freshnessValueId	Comment	Identifier of the freshness
		Type	uint16
		Variation	–
		Direction	IN
Possible Errors	E_OK	Operation successful	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77177: [SecOC] How to convert the secure counter(from uint64 value to uint8 array) in SecOC_GetTxFreshness

Problem description:

In SecOC specification, the SecOC_GetTxFreshness shall be used to obtain the current freshness value (CFV) from Csm. Therefore, the SecOC_GetTxFreshness shall call the Csm_SecureCounterRead function to obtain the CFV (e.g., the maximum length of the CFV is 64bits). However, the SecOC specification is not considered that how to pack the CFV to output parameter. More specifically, I cannot understand how to convert uint64 value (the secure counter value from the Csm_SecureCounterRead) to uint8 array parameter of the SecOC_GetFreshness (SecOCFreshnessValue).

Could you please check and correct it?

Agreed solution:

Add the description in [SWS_SecOC_91002, operation getTXfreshness] to Description:

"Returns the freshness value from the Most Significant Bits in the first byte in the array (SecOCFreshnessValue), in big endian format."

Add the description in [SWS_SecOC_91004] to Description:

"This API returns the freshness value from the Most Significant Bits in the first byte in the array (SecOCFreshnessValue), in big endian format."
–Last change on issue 77177 comment 14–

BW-C-Level:

Application	Specification	Bus
4	4	1

1.40 Specification Item SWS_SecOC_91003

Trace References:

SRS_SECOC_00003, SRS_SECOC_00006

Content:

Service name:	SecOC_GetTxFreshnessTruncDataSecOC_GetTxFreshnessTruncData

Syntax:	Std_ReturnType SecOC_GetTxFreshnessTruncData(uint16 SecOCFreshnessValueID, uint8* SecOCFreshnessValue, uint32* SecOCFreshnessValueLength, uint8* SecOCTruncatedFreshnessValue, uint32* SecOCTruncatedFreshnessValueLength)	
Service ID[hex]:	0x51	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	SecOCFreshnessValueIDSecOC_GetTxFreshnessTruncData.SecOCFreshnessValueID	Holds the identifier of the freshness value.
Parameters (inout):	SecOCFreshnessValueLengthSecOC_GetTxFreshnessTruncData.SecOCFreshnessValueLength	Holds the length of the provided freshness in bits.
	SecOCTruncatedFreshnessValueLengthSecOC_GetTxFreshnessTruncData.SecOCTruncatedFreshnessValueLength	Holds Provides the truncated freshness to be included into the Secured I-PDU. The parameter is optionallength configured for this freshness. The function may adapt the value if needed or can leave it unchanged if the configured length and provided length is the same.
Parameters (out):	SecOCFreshnessValueSecOC_GetTxFreshnessTruncData.SecOCFreshnessValue	Holds the current freshness value.
	SecOCTruncatedFreshnessValueLengthSecOC_GetTxFreshnessTruncData.SecOCTruncatedFreshnessValue Length	Provides Holds the truncated freshness length configured for this freshness. The function may adapt the value if needed or can leave it unchanged if the configured length and provided length is the sameto be included into the Secured I-PDU. The parameter is optional.
Return value:	Std_ReturnType	E_OK: request successful E_NOT_OK: request failed, a freshness value cannot be provided due to general issues for freshness or this FreshnessValueId. E_BUSY: The freshness information can temporarily not be provided.
Description:	This interface is used by the SecOC to obtain the current freshness value. The interface function provides also the truncated freshness transmitted in the secured I-PDU.	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76732: In/Out parameters for function SecOC_GetTxFreshnessTruncData are incorrect

  Problem description:

  Name: WP-x-SEC

Description/Motivation:

In [SWS_SECOC_91003] the parameter SecOCTruncatedFreshnessValueLength is defined as an out parameter, but this should be an InOut parameter. SecOCTruncatedFreshnessValue is defined as InOut parameter but this should be an out parameter.

Agreed solution:

Change [SWS_SECOC_91003]:
Parameters (InOut): SecOCTruncatedFreshnessValueLength.
Parameters (Out): SecOCTruncatedFreshnessValue
–Last change on issue 76732 comment 14–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.41 Specification Item SWS_SecOC_91004

Trace References:

SRS_SECOC_00003, SRS_SECOC_00006

Content:

Service name:	SecOC_GetTxFreshnessSecOC_GetTxFreshness	
Syntax:	Std_ReturnType SecOC_GetTxFreshness(uint16 SecOCFreshnessValueID, uint8* SecOCFreshnessValue, uint32* SecOCFreshnessValueLength)	
Service ID[hex]:	0x52	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	SecOCFreshnessValueIDSecOC_GetTxFreshness.SecOCFreshnessValueID	Holds the identifier of the freshness value.
Parameters (inout):	SecOCFreshnessValueLengthSecOC_GetTxFreshness.SecOCFreshnessValueLength	Holds the length of the provided freshness in bits.
Parameters (out):	SecOCFreshnessValueSecOC_GetTxFreshness.SecOCFreshnessValue	Holds the current freshness value

Return value:	Std_ReturnType	E_OK: request successful E_NOT_OK: request failed, a freshness value cannot be provided due to general issues for freshness or this FreshnessValueId. E_BUSY: The freshness information can temporarily not be provided.
Description:	This interface is used by the SecOC to obtain the current freshness value API returns the freshness value from the Most Significant Bits in the first byte in the array (SecOCFreshnessValue), in big endian format.	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77177: [SecOC] How to convert the secure counter(from uint64 value to uint8 array) in SecOC_GetTxFreshness

  Problem description:

  In SecOC specification, the SecOC_GetTxFreshness shall be used to obtain the current freshness value (CFV) from Csm. Therefore, the SecOC_GetTxFreshness shall call the Csm_SecureCounterRead function to obtain the CFV (e.g., the maximum length of the CFV is 64bits). However, the SecOC specification is not considered that how to pack the CFV to output parameter. More specifically, I cannot understand how to convert uint64 value (the secure counter value from the Csm_SecureCounterRead) to uint8 array parameter of the SecOC_GetFreshness (SecOCFreshnessValue).

  Could you please check and correct it?

  Agreed solution:

  Add the description in [SWS_SecOC_91002, operation getTXfreshness] to Description:

  "Returns the freshness value from the Most Significant Bits in the first byte in the array (SecOCFreshnessValue), in big endian format."

  Add the description in [SWS_SecOC_91004] to Description:

  "This API returns the freshness value from the Most Significant Bits in the first byte in the array (SecOCFreshnessValue), in big endian format."
  –Last change on issue 77177 comment 14–

  BW-C-Level:

Application	Specification	Bus
4	4	1

1.42 Specification Item UC_SecOC_00202

Trace References:

none

Content:

Construction of Freshness value from decoupled counters.

The Freshness Value Manager (FVM) (SW-C or CDD) shall CDD) provide the Freshness Value (FV) to SecOC. FVM shall support a master / slave supports a master-slave synchronization mechanism for the FV.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77782: [SecOC] enhancement of chap.11 (Annex A) to cover new JasPar usecases

  Problem description:

  In R4.3.0 document "Specification of Module Secure Onboard Communication",
  Chapter.11 Annex A, [UC_SecOC_00202] represents the security profile of
  Freshness Value Manager (SW-C) by JasPar.

  Since the current [UC_SecOC_00202] only considers the normal situation,
  JasPar wants to also cover the transient / erroneous situations
  (e.g. in case of message loss, ECU IG-ON, etc.).

  More details of updated points are written below.

  ** Since this UC is referred by many Japanese OEMs and suppliers,
  wed like to update this UC in R4.3.1.
  Please note that this is an UC in Annex A, and therefore will not
  affect the rest of the documents or any other documents.

  ** Note: We will prepare the detailed PS later on for SRS / SWS.

  # 1 FV may be not synchronized between sender and receiver, as a result, MAC error occurs in several cases

  ---

  * Case No.1
  - A. Synchronization message is not received either by sender, or a secured message is not received by receiver,

- B. Multiple synchronization messages are received before next secured message is received by receiver, AND
- C. In the condition that the count up period of reset counter is shorter than the transmission period of secured message.

* Proposal: Define the reset flag at LSB of reset counter in order to detect synchronization error of FV.

* (Case No. 2: this number has been intentionally skipped, to avoid mixing up with # 2.)

* Case No. 3
- In case of sender and receiver which communicate on IG-OFF stats, and when synchronization message is received before message is transmitted in sender, and message is received before synchronization message is received in receiver after IG turns ON

* Proposal: Modify the count up condition of trip counter and reset counter in order to detect synchronization error of FV.

* Case No. 4
- When trip counter and reset counter expire.

* Proposal: Add the definition of expiration behavior of trip counter and reset counter.

# 2 Extension to support the multi-master systems

Details will follow as PS. Only within chap. 11 (Annex A).

Agreed solution:

Refer to the attachment 4585 (Solution Proposal for RfC 77782 - Jaspar Usecase maintenance 5(V0.61))
–Last change on issue 77782 comment 24–

BW-C-Level:

Application	Specification	Bus
1	1	1