Document Title	Specification of Secure Onboard Communication Protocol
Document Category	Protocol specification (PRS)
Document UID	969
Document Status	published
Part of AUTOSAR Standard	Foundation
Part of Standard Release	R26-11

Document Change History			
Date	**Release**	**Changed by**	**Description**
2026-11-26	R26-11	AUTOSAR Release Management	• Update traceable links • Added new spec items related SecOC encryption feature
2025-11-27	R25-11	AUTOSAR Release Management	• Update traceable links
2024-11-27	R24-11	AUTOSAR Release Management	• Removal of implementation specific contents • Removal of configuration parameters not used in the document and not relavant for PRS. • Added new spec items and refined the existing spec items according to the PRS relevance.
2023-11-23	R23-11	AUTOSAR Release Management	• Removal of implementation specific contents • Editorial changes
2022-11-24	R22-11	AUTOSAR Release Management	• Removal of implementation specific contents • Update of configuration parameters • Editorial changes
2021-11-25	R21-11	AUTOSAR Release Management	• no content changes
2020-11-30	R20-11	AUTOSAR Release Management	• Initial release

Disclaimer

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

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

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

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

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Table of Contents

1 Introduction and overview 5
1.1 Protocol purpose and objectives . . . 5
1.2 Applicability of the protocol . . . 6
1.2.1 Constraints and assumptions . . . 6
1.2.1.1 Adaptation in case of asymmetric approach . . . 6
1.2.1.2 Considerations for the AEAD approach . . . 6
1.2.2 Limitations . . . 7
1.3 Dependencies . . . 7
1.3.1 Dependencies to other protocol layers . . . 7
1.3.2 Dependencies to other standards and norms . . . 8
1.3.3 Dependencies to the Application Layer . . . 8
2 Use Cases 9
3 Protocol Requirements 10
3.1 Requirements Traceability . . . 10
4 Definition of terms and acronyms 11
4.1 Acronyms and abbreviations . . . 11
4.2 Definition of terms . . . 11
5 Protocol specification 14
5.1 Specification of the security solution . . . 14
5.1.1 Basic entities of the security solution . . . 14
5.1.2 Authentication of I-PDUs . . . 18
5.1.3 Verification of I-PDUs . . . 20
5.1.3.1 Successful verification of I-PDUs . . . 21
5.1.3.2 Error handling and discarding of reception . . . 21
5.2 Error detection . . . 21
5.3 Security Profiles . . . 22
5.3.1 Overview of security profiles . . . 22
5.3.2 SecOC Profile 1 (or 24Bit-CMAC-8Bit-FV) . . . 22
5.3.3 SecOC Profile 2 (or 24Bit-CMAC-No-FV) . . . 22
5.3.4 SecOC Profile 3 (or JASPAR) . . . 23
5.3.5 SecOC Profile 4 (or AEAD) . . . 23
6 Configuration parameters 25
7 Protocol usage and guidelines 28
8 References 29
A Change history of AUTOSAR traceable items 30
A.1 Change History of this document according to AUTOSAR Release R26-11 30
A.1.1 Added Specification Items in R26-11 . . . 30

A.1.2 Changed Specification Items in R26-11 30
A.1.3 Deleted Specification Items in R26-11 30
A.2 Change History of this document according to AUTOSAR Release R25-11 30
A.2.1 Added Specification Items in R25-11 30
A.2.2 Changed Specification Items in R25-11 30
A.2.3 Deleted Specification Items in R25-11 30
A.3 Change History of this document according to AUTOSAR Release R24-11 31
A.3.1 Added Specification Items in R24-11 31
A.3.2 Changed Specification Items in R24-11 31
A.3.3 Deleted Specification Items in R24-11 31

1 Introduction and overview

Authentication and integrity protection of sensitive data is necessary to protect correct and safe functionality of the vehicle systems - this ensures that received data comes from the right ECU and has the correct value.

The SecOC protocol as described in this document provides a mechanism to verify the authenticity and freshness of PDU based communication between ECUs within the vehicle architecture. The approach requires both the sending ECU and the receiving ECU to implement a SecOC module.

On the sender side, the SecOC module creates a Secured I-PDU by adding authentication information to the outgoing Authentic I-PDU. The authentication information comprises of an Authenticator (e.g. Message Authentication Code) and optionally a Freshness Value. Regardless if the Freshness Value is or is not included in the Secure I-PDU payload, the Freshness Value is considered during generation of the Authenticator. On the receiver side, the SecOC module checks the freshness and authenticity of the Authentic I-PDU by verifying the authentication information that has been appended by the sending side SecOC module. To verify the authenticity and freshness of an Authentic I-PDU, the Secured I-PDU provided to the receiving side SecOC should be the same Secured I-PDU provided by the sending side SecOC and the receiving side SecOC should have knowledge of the Freshness Value used by the sending side SecOC during creation of the Authenticator. Optionally, also confidentiality can be protected by selecting Authenticated Encryption with Associated Data (AEAD) per PDU. A Confidential I-PDU is an Authentic I-PDU that requires additional protection against unauthorized disclosure of its payload. Along with creating the Authenticator, the plaintext is encrypted to a ciphertext to be included in the transmitted Encrypted I-PDU. On receiver side, the Encrypted I-PDU is parsed, and in addition to the authenticity and freshness check, decryption of the payload is performed. Note that depending on the selected algorithms, the overhead may increase and not every bus technology may provide sufficient payload sizes.

1.1 Protocol purpose and objectives

The SecOC protocol aims for resource-efficient and appropriate authentication mechanisms for critical data on the level of PDUs. The authentication mechanisms shall be seamlessly integrated with the current AUTOSAR communication systems. The impact with respect to resource consumption should be as small as possible in order to allow protection as add-on for legacy systems. The specification is based on the assumption that mainly symmetric authentication approaches with message authentication codes (MACs) are used. They achieve the same level of security with much smaller keys than asymmetric approaches and can be implemented compactly and efficiently in software and in hardware. However, the specification provides the necessary level of abstraction so that both, symmetric approaches as well as asymmetric authentication approaches can be used.

1.2 Applicability of the protocol

The SecOC protocol is used in all ECUs where secure communication is necessary.

1.2.1 Constraints and assumptions

1.2.1.1 Adaptation in case of asymmetric approach

Although this document consequently uses the terms and concepts from symmetric cryptography, the SecOC protocol supports both symmetric and asymmetric cryptographic algorithms. In case of an asymmetric approach using digital signatures instead of the MAC-approach described throughout the whole document, some adaptations must be made:

1. Instead of a shared secret between sender and (all) receivers, a key pair consisting of public key and secret key is used. The secret (or private) key is used by the sender to generate the signature, the corresponding public keys is used by (all) receiver(s) to verify the signature. The private key must not be feasibly computable from the public key and it shall not be assessable by the receivers.
2. In order to verify a message, the receiver needs access to the complete signature / output of the signature generation algorithm. Therefore, a truncation of the signature as proposed in the MAC case is NOT possible. The parameter SecOCAuthInfoTruncLength has to be set to the complete length of the signature.
3. The signature verification uses a different algorithm then the signature generation. So instead of "rebuilding" the MAC on receiver side and comparing it with the received (truncated) MAC as given above, the receiver / verifier performs the verification algorithm using the DataToAuthenticator (including full counter) and the signature as inputs and getting a Boolean value as output, determining whether the verification passed or failed.

1.2.1.2 Considerations for the AEAD approach

The option to use Authenticated Encryption with Associated Data (AEAD) instead of Message Authentication Codes (MAC) comes with additional aspects to be considered:

1. A widely used algorithm for AEAD is AES-GCM. This algorithm, however, relies on the uniqueness of Nonces. Nonce reuse may lead to cryptographically catastrophic events such as revealing the secret key stream. In embedded systems, Nonce reuse cannot easily be excluded, as unexpected reboots or failures in persistent storage may happen at any time. Random values, however, cannot guarantee freshness of communication. AES-GCM-SIV was developed to overcome this risk. It may be considered as a valuable alternative for embedded use cases.

2. Nonces and Freshness Value serve the same purpose and may be used interchangeably. To avoid the effects of Nonce reuse, the requirements on Nonces may be stricter than to Freshness Values. Therefore, the APIs are separated to allow implementation of different concepts behind. The handling within SecOC does not vary, once obtained from the respective API.
3. Depending on the algorithm used, the Nonce and/or Authenticator need to be available on receiver side in full length. Whereas the Nonce may be constructed (just as Freshness Values), the Authenticator may be required to be transmitted in full length. This explicitly holds for AES-GCM-SIV.
4. Although AES-GCM and AES-GCM-SIV implement stream ciphers and thus do not prolong the ciphertext in comparison to the plaintext, the overall required size for an Encrypted I-PDU exceeds those of authentication-only Secured I-PDUs. Therefore, the confidentiality option may not be applicable to communication protocols with very limited payload sizes, or additional transport protocols need to be used.
5. Due to the fact that only ciphertext is transmitted, verification status override or skipping verification may not be used, as no useful data will be provided to upper layers.

1.2.2 Limitations

The protocol specification aims to ensure compatibility between AP and CP, and it assumes the communication is realized over ethernet.

Depending of the communication paradigm between AP and CP, the functionality of the protocol is limited. In the case of SOME/IP, the protocol will not support separate transmission of Authentic PDU and Cryptographic PDU and will not support usage of part of the payload as freshness information. (the details are described in the chapter Configuration Parameters.)

1.3 Dependencies

1.3.1 Dependencies to other protocol layers

The interaction of SecOC with the lower layer of the communication stack will depend on the on the platform architecture (AP or CP), and in the case of a CP implementation, it will also depend on the type of transmission: direct transmission, triggered transmission or transport protocol. These design specific dependencies are not part of the protocol specification.

1.3.2 Dependencies to other standards and norms

[1] *IEC 7498-1 The Basic Model, IEC Norm, 1994*

[2,] *National Institute of Standards and Technology (NIST): FIPS-180-4, Secure Hash Standard (SHS), March 2012*, available electronically at `http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf`

[3] *FIPS Pub 197: Advanced Encryption Standard (AES), U.S. Department of Commerce, Information Technology Laboratory (ITL), National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, Federal Information Processing Standards Publication, 2001*, electronically available at `http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf`

1.3.3 Dependencies to the Application Layer

The SecOC protocol does not have dependencies to typical Automotive application. However, it relies on the existence of a software component that provides a freshness information. In addition, there could also be specialized applications that trigger a modification in SecOC behavior (e.g. for development purpose) or applications that monitor the verification results.

2 Use Cases

ID	*Name*	*Description*
UC_001	SecOC SOME/IP	Secure communication between AP and CP using SOME/IP
UC_002	SecOC SignalBased	Secure communication between AP and CP using signal-based communication and SignalToService translation

Table 2.1: Usecases for SecOC protocol

3 Protocol Requirements

3.1 Requirements Traceability

Requirement	Description	Satisfied by
No upstream requirements specified.		

Table 3.1: Requirements Tracing

4 Definition of terms and acronyms

4.1 Acronyms and abbreviations

Abbreviation / Acronym:	Description:
AEAD	Authenticated Encryption with Associated Data
AES	Advanced Encryption Standard
CMAC	Cipher-based Message Authentication Code
FM	Freshness Manager
FV	Freshness Value
GCM-SIV	Galios/Counter Mode - Synthetic Initialization Vector
MAC	Message Authentication Code
Nonce	Number used once
SecOC	Secure Onboard Communication

Table 4.1: Acronyms and abbreviations

4.2 Definition of terms

Terms:	Description:
Authentic I-PDU	An Authentic I-PDU is an arbitrary AUTOSAR I-PDU the content of which is secured during network transmission by means of the Secured I-PDU. The secured content comprises the complete I-PDU or a part of the I-PDU.
Authentication	Authentication is a service related to identification. This function applies to both entities and information itself. Two parties entering into a communication should identify each other. Information delivered over a channel should be authenticated as to origin, date of origin, data content, time sent, etc. For these reasons, this aspect of cryptography is usually subdivided into two major classes: entity authentication and data origin authentication. Data origin authentication implicitly provides data integrity (for if a message is modified, the source has changed).
Authentication Information	The Authentication Information consists of a Freshness Value (or a part thereof) and an Authenticator (or a part thereof). Authentication Information are the additional pieces of information that are added by SecOC to realize the Secured I-PDU.
Authenticator	Authenticator is data that is used to provide message authentication. In general, the term Message Authentication Code (MAC) is used for symmetric approaches while the term Signature or Digital Signature refers to asymmetric approaches having different properties and constraints.
Ciphertext	The result of encrypting a plaintext using a cipher. In the context of this document, it is the encrypted Confidential I-PDU as part of the transmitted Encrypted I-PDU, one of the results of the AEAD Encryption and one input into the AEAD Decryption.
Confidential I-PDU	An Authentic I-PDU that requires additional protection against unauthorized disclosure of its payload and thus is intended to be encrypted before transmission.

Terms:	Description:
Data integrity	Data integrity is the property whereby data has not been altered in an unauthorized manner since the time it was created, transmitted, or stored by an authorized source. To assure data integrity, one should have the ability to detect data manipulation by unauthorized parties. Data manipulation includes such things as insertion, deletion, and substitution.
Data origin authentication	Data origin authentication is a type of authentication whereby a party is corroborated as the (original) source of specified data created at some (typically unspecified) time in the past. By definition, data origin authentication includes data integrity.
Distinction unilateral / bilateral authentication	In unilateral authentication, one side proves identity. The requesting side is not even authenticated to the extent of proving that it is allowed to request authentication. In bilateral authentication, the requester is also authenticated at least (see below) to prove the privilege of requesting. There is an efficient and more secure way to authenticate both endpoints, based on the bilateral authentication described above. Along with the authentication (in the second message) requested initially by the receiver (in the first message), the sender also requests an authentication. The receiver sends a third message providing the authentication requested by the sender. This is only three messages (in contrast to four with two unilateral messages).
Encrypted I-PDU	A Secured I-PDU that provides additional protection against unauthorized disclosure of its payload as it gets encrypted before transmission.
Entity authentication	Entity authentication is the process whereby one party is assured (through acquisition of corroborative evidence) of the identity of a second party involved in a protocol, and that the second has actually participated (i.e., is active at, or immediately prior to, the time the evidence is acquired). **Note:** Entity authentication means to prove presence and operational readiness of a communication endpoint. This is for example often done by proving access to a cryptographic key and knowledge of a secret. It is necessary to do this without disclosing either key or secret. Entity authentication can be used to prevent record-and-replay attacks. Freshness of messages only complicates them by the need to record a lifetime and corrupt either senders or receivers (real-time) clock. Entity authentication is triggered by the receiver, i.e. the one to be convinced, while the sender has to react by convincing. Record and replay attacks on entity authentication are usually prevented by allowing the receiver some control over the authentication process. In order to prevent the receiver from using this control for steering the sender to malicious purposes or from determining a key or a secret ("oracle attack"), the sender can add more randomness. If not only access to a key (implying membership to a privileged group) but also individuality is to be proven, the sender additionally adds and authenticates its unique identification.

Terms:	Description:
Message authentication	Message authentication is a term used analogously with data origin authentication. It provides data origin authentication with respect to the original message source (and data integrity, but no uniqueness and timeliness guarantees).
Secured I-PDU	A Secured I-PDU is an AUTOSAR I-PDU that contains Payload of an Authentic I-PDU supplemented by additional Authentication Information.
Transaction authentication	Transaction authentication denotes message authentication augmented to additionally provide uniqueness and timeliness guarantees on data (thus preventing undetectable message replay).

Table 4.2: Definition of terms

5 Protocol specification

5.1 Specification of the security solution

The SecOC protocol as described in this document provides a mechanism to verify the authenticity and freshness of PDU based communication between ECUs within the vehicle architecture. The approach requires both the sending ECU and the receiving ECU to implement a SecOC module.

On the sender side, the SecOC module creates a Secured I-PDU by adding Authentication Information to the outgoing Authentic I-PDU. The Authentication Information comprises of an Authenticator (e.g. Message Authentication Code) and optionally a Freshness Value. Regardless if the Freshness Value is or is not included in the Secure I-PDU payload, the Freshness Value is considered during generation of the Authenticator.

On the receiver side, the SecOC module checks the freshness and authenticity of the Authentic I-PDU by verifying the Authentication Information that has been appended by the sending side SecOC module. To verify the authenticity and freshness of an Authentic I-PDU, the Secured I-PDU provided to the receiving side SecOC should be the same Secured I-PDU provided by the sending side SecOC and the receiving side SecOC should have knowledge of the Freshness Value used by the sending side SecOC during creation of the Authenticator.

5.1.1 Basic entities of the security solution

The term Authentic I-PDU refers to an AUTOSAR I-PDU that requires protection against unauthorized manipulation and replay attacks. A Confidential I-PDU is an Authentic I-PDU that requires additional protection against unauthorized disclosure of its payload and thus is intended to be encrypted before transmission.

The payload of a Secured I-PDU consists of the Authentic I-PDU and an Authenticator (e.g. Message Authentication Code). The payload of a Secured I-PDU may optionally include the Freshness Value used to create the Authenticator (e.g. MAC). An Encrypted I-PDU is a Secured I-PDU that provides additional protection against unauthorized disclosure of its payload as it gets encrypted before transmission. In there, the Authentic I-PDU is replaced by the Ciphertext, i.e. the encrypted Confidential I-PDU [4]. The order in which the contents are structured in the Secured I-PDU is compliant with Figure 5.1 or Figure 5.2, respectively.

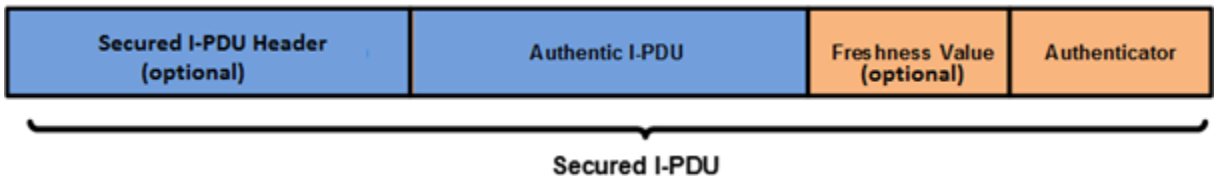

Figure 5.1: Secured I-PDU contents

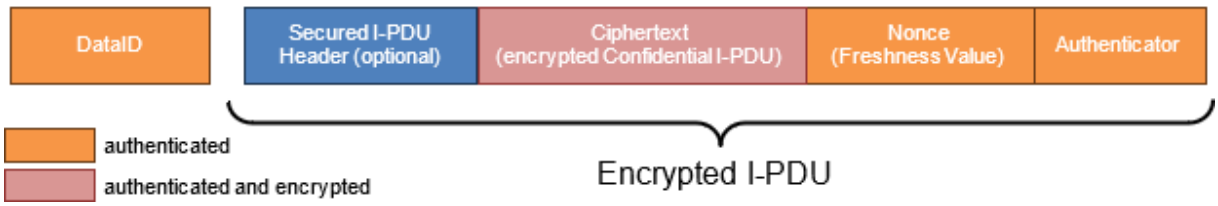

Figure 5.2: `Encrypted I-PDU` contents

The length of the `Authentic I-PDU`, the Freshness Value and the `Authenticator` within a `Secured I-PDU` may vary from one uniquely indefinable `Secured I-PDU` to another.

The `Authenticator` (e.g. `MAC`) refers to a unique authentication data string generated using a Key, Data Identifier of the `Secured I-PDU`, Authentic Payload, and Freshness Value. The `Authenticator` provides a high level of confidence that the data in an `Authentic I-PDU` is generated by a legitimate source and is provided to the receiving ECU at the time in which it is intended for.

Depending on the authentication algorithm used to generate the `Authenticator`, it may be possible to truncate the resulting `Authenticator` (e.g. in case of a `MAC`) generated by the authentication algorithm. Truncation may be desired when the message payload is limited in length and does not have sufficient space to include the full `Authenticator`.

The `Authenticator` length contained in a `Secured I-PDU` (parameter `SecOCAuthInfoTruncLength`) is specific to a uniquely identifiable `Secured I-PDU`. This allows provision of flexibility across the system (i.e. two independent unique Secured I-PDUs may have different `Authenticator` lengths included in the payload of the `Secured I-PDU`) by providing fine grain configuration of the `MAC` truncation length for each `Secured I-PDU`.

If truncation is possible, the `Authenticator` should only be truncated down to the most significant bits of the resulting `Authenticator` generated by the authentication algorithm. Figure 5.3 shows an example of the truncation of the `Authenticator` and the Freshness Values respecting the parameter `SecOCFreshnessValueTruncLength` and `SecOCAuthInfoTruncLength`.

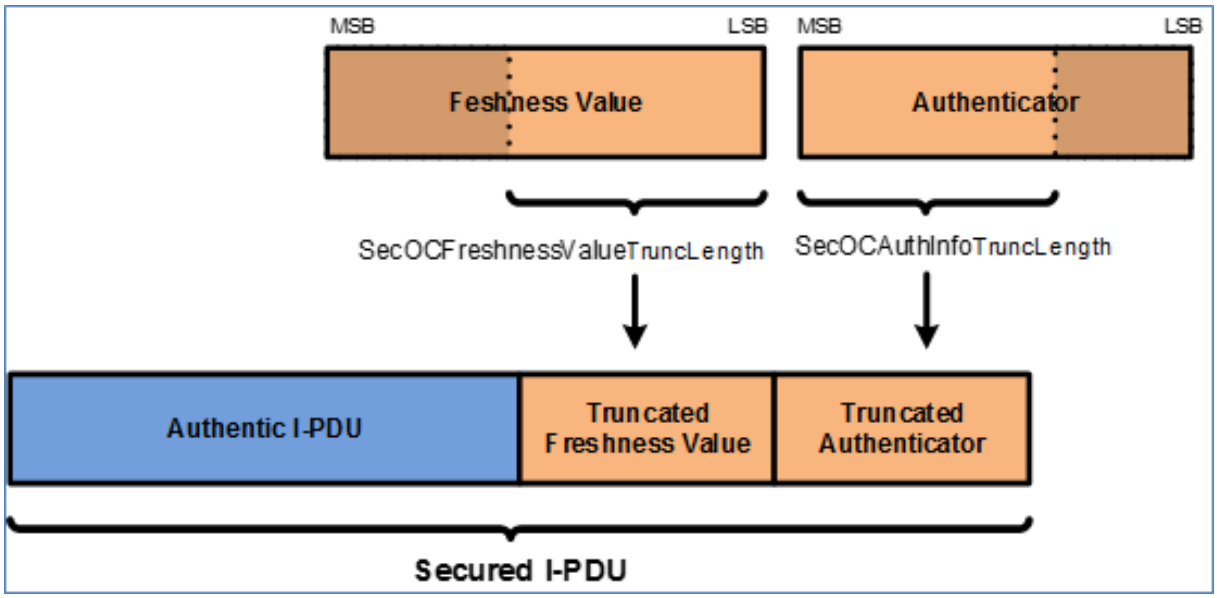

Figure 5.3: An example of `Secured I-PDU` contents with truncated `Freshness Counter` and truncated `Authenticator` (without `Secured I-PDU` Header)

Note: For the resource constraint embedded use case with static participants, we propose using Message Authentication Codes (MACs) as a basis for authentication (e.g. a CMAC [5] based on AES [3] with an adequate key length).

Note: In case a `MAC` is used, it is possible to transmit and compare only parts of the `MAC`. This is known as `MAC` truncation. However, this results in a lower security level at least for forgery of single MACs. While we propose to always use a key length of at least 128 bits, a `MAC` truncation can be beneficial. Of course, the actual length of the `MAC` for each use case has to be chosen carefully. For some guidance, we refer to appendix A of [5]. In general, `MAC` sizes of 64 bit and above are considered to provide sufficient protection against guessing attacks by NIST. Depending on the use case, different `MAC` sizes can be appropriate, but this requires careful judgment by a security expert.

[PRS_SecOc_00125] Secure I-PDU construction ⌈Starting from the first byte, the payload of a `Secured I-PDU` shall consist of:

1. The `Secured I-PDU` Header(optional) - This is an optional parameter, that indicates the length of the `Authentic I-PDU` in bytes.
2. The `Authentic I-PDU`.
3. The Freshness Value (optional) used to create the `Authenticator`.
4. The `Authenticator`(e.g. Message Authentication Code).

⌋

[PRS_SecOc_00101] ⌈All `SecOC` data (e.g. Freshness Value, `Authenticator`, Data Identifier, `SecOC` message link data,...) that is directly or indirectly transmitted to the

other side of a communication link shall be encoded in Big Endian byte order so that the data is interpreted in the same way on reception.⌋

[PRS_SecOc_00102] ⌈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.⌋

Each Secured I-PDU is configured with at least one Freshness Value. The Freshness Value refers to a monotonic counter that is used to ensure freshness of the Secured I-PDU. Such a monotonic counter could be realized by means of individual message counters, called Freshness Counter, or by a time stamp value called Freshness Timestamp.

[PRS_SecOc_00126] SecOCAuthPduHeaderLength specification ⌈The parameter SecOCAuthPduHeaderLength specifies the length of the Secured I-PDU Header from 0 to 4 bytes.⌋

[PRS_SecOc_00103] ⌈If the parameter SecOCFreshnessValueTruncLength is configured to a smaller length than the actual Freshness Value, SecOC shall include only the least significant bits of the freshness value up to SecOCFreshnessValueTruncLength within the Secured I-PDU.
If the parameter SecOCFreshnessValueTruncLength is configured to 0, the Freshness Value shall not be included in the Secured I-PDU.⌋

[PRS_SecOc_00104] ⌈If SecOCUseAuthDataFreshness is set to TRUE, SecOC shall use a part of the Authentic I-PDU for freshness generation and verification. In this case, SecOCAuthDataFreshnessStartPosition determines the start position in bits of the freshness inside the Authentic I-PDU and SecOCAuthDataFreshnessLen determines its length in bits.⌋

Note: This allows reusing existing Freshness Values from the payload which are guaranteed to be unique within the validity period of a Freshness Timestamp, e.g. a 4-bit E2E counter. In this case SecOC does not need to generate any additional counter values.

[PRS_SecOc_00105] ⌈The Freshness Value shall be aligned to the MSB of the first byte in the array. The 15th bit of the Freshness Value is the MSB of the 2nd byte and so on. Unused bits of the freshness array shall be set to 0. The associated length information shall be given in bits.⌋

[PRS_SecOc_00127] Padding bytes specification ⌈The padding bytes shall be set to the value configured by the parameter SecOCPduPaddingbytes.⌋

5.1.2 Authentication of I-PDUs

[PRS_SecOc_00200] ⌈The creation of a Secured I-PDU consists of the following steps in this order:

1. Generate Authenticator or perform Authenticated Encryption.
2. Construct Secured I-PDU or Encrypted I-PDU.
3. Increment Freshness Counter.

It shall be ensured that the Authenticator is generated before the freshness counter is incremented.⌋

[PRS_SecOc_00224] AEAD Encryption Inputs and Outputs ⌈The data provided as input into AEAD encrypt algorithm consists of the SecOCDataId as Associated Data, the Confidential I-PDU as plaintext and the Freshness Value/Freshness Nonce as the initialization vector. Its outputs are the Ciphertext and the Authenticator.⌋

[PRS_SecOc_00206] ⌈If the Freshness value calculation or the authenticator calculation fails and a default Pattern is configured by parameter SecOCDefaultAuthenticationInformationPattern, then Secured I-PDU shall use the default pattern for all the bytes of Freshness Value and Authenticator.⌋

[PRS_SecOc_00221] ⌈If the Freshness value calculation or the authenticator calculation fails and a default Pattern is not configured by parameter SecOCDefaultAuthenticationInformationPattern, then Secured I-PDU shall be dropped.⌋

Note:
Example:
SecOCFreshnessValueTxLength = 4bits
SecOCAuthInfoTxLength = 20 bits
SecOCDefaultAuthenticationInformationPattern = 0xA5
The resulting default Authentication Information within the secured PDU would be 0x05 (Truncated Freshness Value) | 0xA5 0xA5 0xA0 (Truncated Authenticator). "|" denotes concatenation.

[PRS_SecOc_00222] ⌈The Data Identifier of the Secured I-PDU (SecOCDataId) has a size of 16-bits.⌋

[PRS_SecOc_00208] ⌈The data, on which the Authenticator is calculated, consists of SecOCDataId, Authentic I-PDU data and Complete Freshness Value in the given order. These are concatenated together respectively to make up the bit array that is passed into the authentication algorithm for Authenticator generation/verification.

DataToAuthenticator = Data Identifier | secured part of the Authentic I-PDU | Complete Freshness Value.⌋

Note: "|" denotes concatenation

[PRS_SecOc_00210] ⌈The Authenticator shall be truncated down to the number of bits specified by the parameter SecOCAuthInfoTruncLength.⌋

[PRS_SecOc_00211] ⌈The Secured I-PDU shall be constructed 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 [SecOCFreshnessValueTruncLength] (optional) | Authenticator [SecOCAuthInfoTruncLength]⌋

[PRS_SecOc_00225] AEAD Secured I-PDU Construction ⌈If AEAD is used, the Secured I-PDU (i.e. Encrypted I-PDU) shall be constructed by adding the Secured I-PDU Header (optional), the Nonce and the Authenticator to the Ciphertext.

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) | Ciphertext | Nonce [SecOCFreshnessValueTruncLength] (optional) | Authenticator [SecOCAuthInfoTruncLength]⌋

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.

[PRS_SecOc_00213] ⌈In case of frame length constraints, the SecOCSecuredPduCollection could be used.

In this case, the Secured I-PDU shall be split to two messages : The original Authentic I-PDU and a separate Cryptographic I-PDU. This Cryptographic I-PDU shall contain all Authentication Information: the authenticator and the Freshness Value.⌋

[PRS_SecOc_00215] ⌈If SecOCSecuredPduCollection is used along with SecOCUseMessageLink, a part of the Authentic I-PDU shall be repeated inside the Cryptographic I-PDU as Message Linker.

In this case, the Cryptographic I-PDU shall be constructed as Cryptographic I-PDU = Authentication Data | Message Linker⌋

Note: "|" denotes concatenation

[PRS_SecOc_00216] ⌈If SecOCUseMessageLink is used then SecOC shall use the value at bit position SecOCMessageLinkPos of length SecOCMessageLinkLen bits inside the Authentic I-PDU as the Message Linker.⌋

[PRS_SecOc_00220] ⌈For a Tx Secured I-PDU with SecOCAuthPduHeader-Length > 0, the Secured I-PDU Header shall denote the length of the Authentic I-PDU within the Secured I-PDU, to handle dynamic Authentic I-PDU.⌋

5.1.3 Verification of I-PDUs

[PRS_SecOc_00300] ⌈The verification of a Secured I-PDU consists of the following 3 steps:

1. Calculate the expected Freshness Value.
2. Construct Data for Authentication or AEAD Verification.
3. Verify Authentication Information or perform AEAD verification.

⌋

[PRS_SecOc_00226] AEAD Decryption Inputs and Outputs ⌈The data provided as input into AEAD decrypt algorithm consists of the SecOCDataId as associated data, the Ciphertext transmitted in the Encrypted I-PDU and the Freshness Value/Freshness Nonce as the initialization vector. Its outputs are the Confidential I-PDU (plaintext) and the verification result.⌋

[PRS_SecOc_00306] ⌈If the verification of the Authenticator could be successfully executed but the verification failed (e.g. the MAC verification has failed or the key was invalid), the authentication verify attempt counter shall be incremented, and the Freshness Value shall be updated to the next valid value.⌋

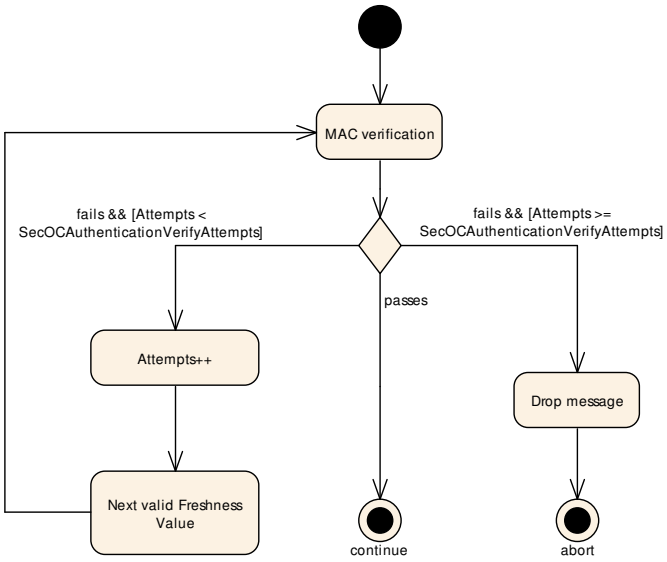

Figure 5.4: Verification of MAC

[PRS_SecOc_00309] ⌈If the authentication verify attempt counter has reached the threshold of the SecOCAuthenticationVerifyAttempts on a failed verify attempt, then the received PDU shall be discarded.⌋

[PRS_SecOc_00223] ⌈If the parameter SecOCEnableForcedPassOverride is true, then the SecOC shall provide a mechanism to allow reception of the PDUs that failed in authentication.⌋

[PRS_SecOc_00316] ⌈The data that is used to calculate the Authenticator (DataToAuthenticator) on the receiver side shall be constucted. This data is comprised of SecOCDataId | AuthenticIPDU | FreshnessVerifyValue.⌋

[PRS_SecOc_00317] ⌈The receiver shall calculate the expected Authenticator by passing DataToAuthenticator, length of DataToAuthenticator and SecOCAuthInfoTruncLength into the authentication algorithm. The expected Authenticator calculated shall be verified against the Authenticator parsed from the Secured I-PDU.⌋

5.1.3.1 Successful verification of I-PDUs

[PRS_SecOc_00320] ⌈If the verification of a Secured I-PDU was successful or the status override was set accordingly, the SecOC module shall pass the Authentic I-PDU to the upper layer communication modules using the lower layer interfaces of the communication stack.⌋

5.1.3.2 Error handling and discarding of reception

[PRS_SecOc_00342] ⌈If SecOC has received both an Authentic I-PDU and a Cryptographic PDU and the verification of the resulting Secured I-PDU fails, both the Authentic and Cryptographic I-PDU shall remain buffered and verification shall be reattempted each time new data for any of them is received.⌋

Note: This and the above requirement ensure that even if either an Authentic I-PDU or a Cryptographic I-PDU is lost in transit, SecOC will still function as expected as soon as an Authentic I-PDU and its corresponding Cryptographic I-PDU are received in direct succession.

5.2 Error detection

This chapter has no content.

5.3 Security Profiles

5.3.1 Overview of security profiles

Secure Onboard Communication protocol allows multiple cryptographic algorithms and modes for the MAC calculation and how the truncation of the MAC and Freshness Value(if applicable) shall be done.The security profiles provide a consistent set of values for a subset of configuration parameters that are relevant for the configuration of Secure Onboard Communication.

[PRS_SecOc_00600] ⌈Each Security Profile shall provide the configuration values for the authentication algorithm (parameter `algorithmFamily`, `algorithmMode` and `algorithmSecondaryFamily` in `CryptoServicePrimitive`), length of freshness Value, if applicable (parameter `SecOCFreshnessValueLength`), length of truncated Freshness Value (parameter `SecOCFreshnessValueTruncLength`), length of truncated MAC (parameter `SecOCAuthInfoTruncLength`), and a description of the profile.⌋

5.3.2 SecOC Profile 1 (or 24Bit-CMAC-8Bit-FV)

[PRS_SecOc_00610] ⌈Using the CMAC algorithm based on AES-128 according to NIST SP 800-38B to calculate the MAC, use the eight least significant bit of the Freshness Value as Truncated Freshness Value and use the 24 most significant bits of the MAC as truncated MAC.⌋

Parameter	Configuration value
The algorithm for the MAC (parameter `algorithmFamily`)	CRYPTO_ALGOFAM_AES
The algorithm mode for the MAC (parameter `algorithmMode`)	CRYPTO_ALGOMODE_CMAC
Additional algorithm family configuration (parameter `algorithmSecondaryFamily`, not used in this profile)	CRYPTO_ALGOFAM_NOT_SET
Length of Freshness Value (parameter `SecOCFreshnessValueLength`)	Not Specified
Length of truncated Freshness Value (parameter `SecOCFreshnessValueTruncLength`)	8 bits
Length of truncated MAC (parameter `SecOCAuthInfoTruncLength`)	24 bits

5.3.3 SecOC Profile 2 (or 24Bit-CMAC-No-FV)

[PRS_SecOc_00620] ⌈Using the CMAC algorithm based on AES-128 according to NIST SP 800-38B to calculate the MAC, don't use any freshness value at all and use the 24 most significant bits of the MAC as truncated MAC. The profile shall only be used if no synchronized freshness value is established. There is no restriction to a special bus.⌋

Parameter	Configuration value
The algorithm for the MAC (parameter algorithmFamily)	CRYPTO_ALGOFAM_AES
The algorithm mode for the MAC (parameter algorithmMode)	CRYPTO_ALGOMODE_CMAC
Additional algorithm family configuration (parameter algorithmSecondaryFamily, not used in this profile)	CRYPTO_ALGOFAM_NOT_SET
Length of Freshness Value (parameter SecOCFreshnessValueLength)	0
Length of truncated Freshness Value (parameter SecOCFreshnessValueTruncLength)	0 bits
Length of truncated MAC (parameter SecOCAuthInfoTruncLength)	24 bits

5.3.4 SecOC Profile 3 (or JASPAR)

[PRS_SecOc_00630] ⌈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_SecOC_00202]. The profile shall be used for CAN.⌋

Parameter	Configuration value
The algorithm for the MAC (parameter algorithmFamily)	CRYPTO_ALGOFAM_AES
The algorithm mode for the MAC (parameter algorithmMode)	CRYPTO_ALGOMODE_CMAC
Additional algorithm family configuration (parameter algorithmSecondaryFamily, not used in this profile)	CRYPTO_ALGOFAM_NOT_SET
Length of Freshness Value (parameter SecOCFreshnessValueLength)	64 bits
Length of truncated Freshness Value (parameter SecOCFreshnessValueTruncLength)	4 bits
Length of truncated MAC (parameter SecOCAuthInfoTruncLength)	28 bits

5.3.5 SecOC Profile 4 (or AEAD)

[PRS_SecOc_00640] AES-GCM-SIV AEAD Profile ⌈This profile depicts one configuration and usage of Authenticated Encryption with Associated Data, based on AES-GCM-SIV, according to RFC 8452. Note that AES-GCM-SIV requires the full authenticator as input into the decryption algorithm. Therefore, truncation of the authenticator is NOT possible. This algorithm is, in contrast to AES-GCM, resilient against Nonce-reuse, which is hardly to be excluded in embedded systems.⌋

Parameter	Configuration value
The algorithm for the MAC (parameter algorithmFamily)	CRYPTO_ALGOFAM_AES
The algorithm mode for the MAC (parameter algorithmMode)	CRYPTO_ALGOMODE_GCM_SIV

Parameter	Configuration value
Additional algorithm family configuration (parameter `algorithmSecondaryFamily`, not used in this profile)	CRYPTO_ALGOFAM_NOT_SET
Length of Nounce (parameter `SecOCFreshnessValueLength`)	96 bits
Length of truncated Nounce (parameter `SecOCFreshnessValueTruncLength`)	96 bits
Length of truncated `MAC` (parameter `SecOCAuthInfoTruncLength`)	128 bits

6 Configuration parameters

The table below describes the configuration parameters for the protocol. For the communication between AP and CP using SOME/IP network binding or raw data streaming, the protocol has a reduced functionality, therefore some parameters are not available in AP or they are implementation specific. These are described in column "Applicability to AP SOME/IP network binding". As a consequence, the requirements referring these parameters are not applicable either. The parameters that are not available are because following features are not available:

- It is not possible to use part of the Authentic PDU to construct as freshness information
- It is not possible to separate the Secure PDU in two different PDUs: Authentic and Cryptographic PDUs
- Provision of Freshness value already truncated by FM (the truncation is always done by SecOC)

The parameters defined in the table below will be referenced in the PRS to represent the configurable parameters or configurability in the SecOC protocol . They are not directly related to the Autosar Classic EcuC or the Autosar Adaptive Manifest.

Parameter	***Description***	***Applicability to AP SOME/IP network binding***
SecOCAuthPduHeaderLength	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.	no
SecOCFreshnessValueTruncLength	This parameter defines the length in bits of the Freshness Value to be included in the payload of the Secured I-PDU. This length is specific to the least significant bits of the complete Freshness Counter. If the parameter is 0 no Freshness Value is included in the Secured I-PDU.	yes

▽

△

SecOCUseAuthDataFreshness	A Boolean value that indicates if a part of the Authentic I-PDU is used for freshness generation and verification. If it is set to TRUE, the values SecOCAuthData-FreshnessStartPosition and SecOCAuthDataFreshnessLen specifies the bit position and length of the Freshness value within the Authentic I-PDU.	no
SecOCAuthDataFreshnessStart-Position	This value determines the start position in bits (uint16) of the Authentic PDU that is used for Freshness Value determination.	no
SecOCAuthDataFreshnessLen	This attribute defines the length in bits of the authentic PDU data for Freshness Value determination.	no
SecOCDefaultAuthenticationInfor-mationPattern	The pattern, configured in this parameter is used as bytes of Freshness Value and Authenticator, if the Freshness value calculation or the Authenticator calculation fails.	Implementation specific
SecOCDataId	This parameter defines a unique numerical identifier for the Secured I-PDU.	yes
SecOCAuthInfoTruncLength	This parameter defines the length in bits of the authentication code to be included in the payload of the Secured I-PDU.	yes
SecOCSecuredPduCollection	The secured I-PDU shall be split to two messages : The original Authentic I-PDU and a separate Cryptographic I-PDU.	no
SecOCFreshnessValueLength	This parameter defines the complete length in bits of the Freshness Value. As long as the key doesn't change the counter shall not overflow. The length of the counter shall be determined based on the expected lifetime of the corresponding key and frequency of usage of the counter.	yes

▽

△

SecOCPduPaddingbytes	The bytes to be padded in a secured PDU is updated with this pattern.	Implementation specific
SecOCUseMessageLink	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.	no
SecOCMessageLinkPos	The position of the Message Linker inside the Authentic I-PDU in bits. The bit counting is done according to 01068 and the bit ordering is done according to TPS_SYST_01069.	no
SecOCMessageLinkLen	Length of the Message Linker inside the Authentic I-PDU in bits.	no
SecOCEnableForcedPassOverride	This parameter defines whether the signature authentication or MAC verification is performed on this Secured I-PDU. If set to false, the Authentic I-PDU is extracted from the Secured I-PDU without verification.	Implementation specific
SecOCAuthenticationVerifyAttempts	This parameter specifies the number of authentication verify attempts that are to be carried out when the verification of the authentication information failed for a given Secured I-PDU. If zero is set, then only one authentication verification attempt is done.	yes
SecOCUseAuthenticatedEncryption	A Boolean value that indicates if the Authentic I-PDU shall be handled as Confidential I-PDU. If set to TRUE, Authenticated Encryption with Associated Data shall be applied. Note that SecOCTxAuthServiceConfigRef needs to be set accordingly.	yes

Table 6.1: Configuration parameters for SecOC protocol

7 Protocol usage and guidelines

This chapter has no content.

8 References

[1] IEC: The Basic Model, IEC Norm

[2] NIST: Secure Hash Standard (SHS)
http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf

[3] NIST: Announcing the Advanced Encryption Standard (AES)
http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[4] AES-GCM-SIV: Nonce Misuse-Resistant Authenticated Encryption
https://datatracker.ietf.org/doc/html/rfc8452

[5] NIST Special Publication 800-38B: Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication
http://csrc.nist.gov/publications/nistpubs/800-38B/SP_800-38B.pdf

A Change history of AUTOSAR traceable items

A.1 Change History of this document according to AUTOSAR Release R26-11

A.1.1 Added Specification Items in R26-11

[PRS_SecOc_00224] [PRS_SecOc_00225] [PRS_SecOc_00226] [PRS_SecOc_00640]

A.1.2 Changed Specification Items in R26-11

[PRS_SecOc_00101] [PRS_SecOc_00102] [PRS_SecOc_00103] [PRS_SecOc_00104] [PRS_SecOc_00105] [PRS_SecOc_00125] [PRS_SecOc_00126] [PRS_SecOc_00127] [PRS_SecOc_00200] [PRS_SecOc_00206] [PRS_SecOc_00208] [PRS_SecOc_00210] [PRS_SecOc_00211] [PRS_SecOc_00213] [PRS_SecOc_00215] [PRS_SecOc_00216] [PRS_SecOc_00220] [PRS_SecOc_00221] [PRS_SecOc_00222] [PRS_SecOc_00223] [PRS_SecOc_00300] [PRS_SecOc_00306] [PRS_SecOc_00309] [PRS_SecOc_00316] [PRS_SecOc_00317] [PRS_SecOc_00320] [PRS_SecOc_00342] [PRS_SecOc_00600] [PRS_SecOc_00610] [PRS_SecOc_00620] [PRS_SecOc_00630]

A.1.3 Deleted Specification Items in R26-11

none

A.2 Change History of this document according to AUTOSAR Release R25-11

A.2.1 Added Specification Items in R25-11

none

A.2.2 Changed Specification Items in R25-11

none

A.2.3 Deleted Specification Items in R25-11

none

A.3 Change History of this document according to AUTOSAR Release R24-11

A.3.1 Added Specification Items in R24-11

[PRS_SecOc_00125] [PRS_SecOc_00126] [PRS_SecOc_00127]

A.3.2 Changed Specification Items in R24-11

none

A.3.3 Deleted Specification Items in R24-11

[PRS_SecOc_00313] [PRS_SecOc_00314] [PRS_SecOc_00315] [PRS_SecOc_00330] [PRS_SecOc_00340] [PRS_SecOc_00341] [PRS_SecOc_00500]