ETSI TS 102 723-8 V1.1.1 (2016-04)

Intelligent Transport Systems (ITS);
OSI cross-layer topics;
Part 8: Interface between security entity and network and transport layer

Reference
DTS/ITS-0050008

Keywords
adaption, addressing, interface, ITS

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Intelligent Transport Systems (ITS).

The present document is part 8 of a multi-part deliverable. Full details of the entire series can be found in part 1 [i.1].

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Introduction

ITS stations are complex systems that may be implemented in different ways. The reference architecture is described in the communications architecture standard ETSI EN 302 665 [1], clause 4.4. The present document aims to address the security interface from a functional point of view. Access control to the Service Access Point and further definitions of station internals are out of scope of the present document.

The SAP specification is specific to the ITS architecture but generic to the concrete technologies used.

Therefore, the present document is structured in the following way:

First, the architecture integration is outlined. Secondly, functionalities are collected from related standards and mapped to service primitives. Finally, the use of service primitives in procedures is described.

1 Scope

The present document specifies interfaces between the ITS security entity and the ITS network and transport layers including interface services and service primitives which are extensible in order to achieve general applicability. Additionally, it specifies related procedures and common parameters.

The SN-SAP description in the present document is functional as according to the ISO model as modified by ETSI EN 302 665 [1].

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference/.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI EN 302 665: "Intelligent Transport Systems (ITS); Communications Architecture".

[2] ETSI TS 102 940: "Intelligent Transport Systems (ITS); Security; ITS communications security architecture and security management".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI TS 102 723-1: "Intelligent Transport Systems (ITS); OSI cross-layer topics; Part 1: Architecture and addressing schemes".

[i.2] ETSI TS 101 539-2: "Intelligent Transport System (ITS); V2X Applications; Intersection Collision Risk Warning (ICRW) application requirements specification".

[i.3] ETSI TS 101 539-3: "Intelligent Transport Systems (ITS); V2X Applications; Part 3: Longitudinal Collision Risk Warning (LCRW) application requirements specification".

[i.4] PRE-DRIVE C2X Deliverable D1.3: "Security Architecture".

[i.5] EVITA Deliverable D3.2: "Secure On-board Architecture Specification".

[i.6] ETSI TS 102 637-1: "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 1: Functional Requirements".

[i.7] ETSI TS 102 637-2: "Intelligent Transport Systems (ITS);Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service".

[i.8] ETSI ES 202 663: "Intelligent Transport Systems (ITS); European profile standard for the physical and medium access control layer of Intelligent Transport Systems operating in the 5 GHz frequency band".

[i.9] ISO 24102-3: "Intelligent transport systems -- Communications access for land mobiles (CALM) -- ITS station management -- Part 3: Service access points".

[i.10] ETSI TS 103 097: "Intelligent Transport Systems (ITS); Security; Security header and certificate formats".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ETSI EN 302 665 [1], ETSI TS 102 940 [2] and the following apply:

security association: addressing information and 'security material' for connecting to the 'security management entity'

NOTE: This corresponds to 'enrolment authorities' and 'authorization authorities'.

security entity: functional entity inside an ITS station which offers 'security mechanisms'

security protocol: protocol used to encode and decode 'security material' and messages between ITS Stations

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI EN 302 665 [1], ETSI TS 102 940 [2] and the following apply:

ASN	Abstract Syntax Notation
CAM	Cooperative Awareness Message
DENM	Decentralized Environmental Notification Message
ICRW	Intersection Collision Risk Warning
ID	'pseudonym' identity
IN-SAP	access layer - networking & transport layer SAP
ISO	International Organization for Standardization
ITS-S	ITS-Station
LCRW	Longitudinal Collision Risk Warning
NF-SAP	Networking & transport layer - Facilities layer SAP
RX	Receiver
SA	Security Association

NOTE: SA is contextual dependent either "name of interface between security entity and ITS-S applications" as given in ETSI EN 302 665 [1] or "Security Association".

SAP	Service Access Point
SF-SAP	Security entity - Facilities layer SAP
SN-SAP	Security entity - Networking & transport layer SAP
TX	Transmitter

4 Architecture integration

4.1 General

4.1.1 Introduction

Figure 1 shows the ITS station reference architecture, as defined in ETSI EN 302 665 [1]. The present document contains the specification of the Service Access Points (SAP), connecting the security entity and the networking and transport layers, i.e. SN-SAP.

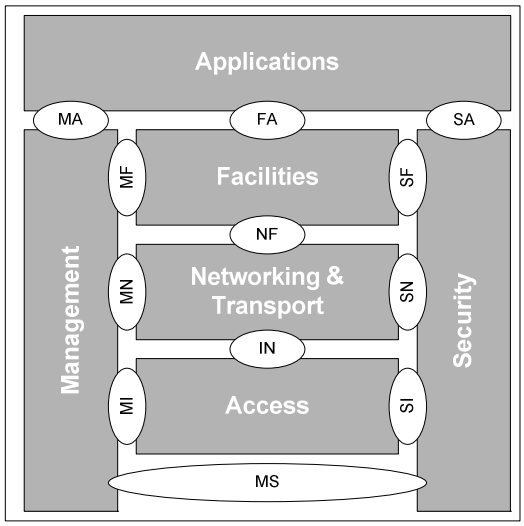

Figure 1: ITS station reference architecture

Interaction between the security entity and the layers may follow two principles. First, the vertical message flow through the layers from top to bottom or vice versa. Secondly, the horizontal control communication from the security entity towards the corresponding layer. Both are described in clauses 4.1.2 and 4.1.3.

4.1.2 Vertical message flow

Figure 2 extends the ITS station reference architecture by illustrating the overall information flow through the layers, from originating application on the left hand side, to the receiving application on the right hand side.

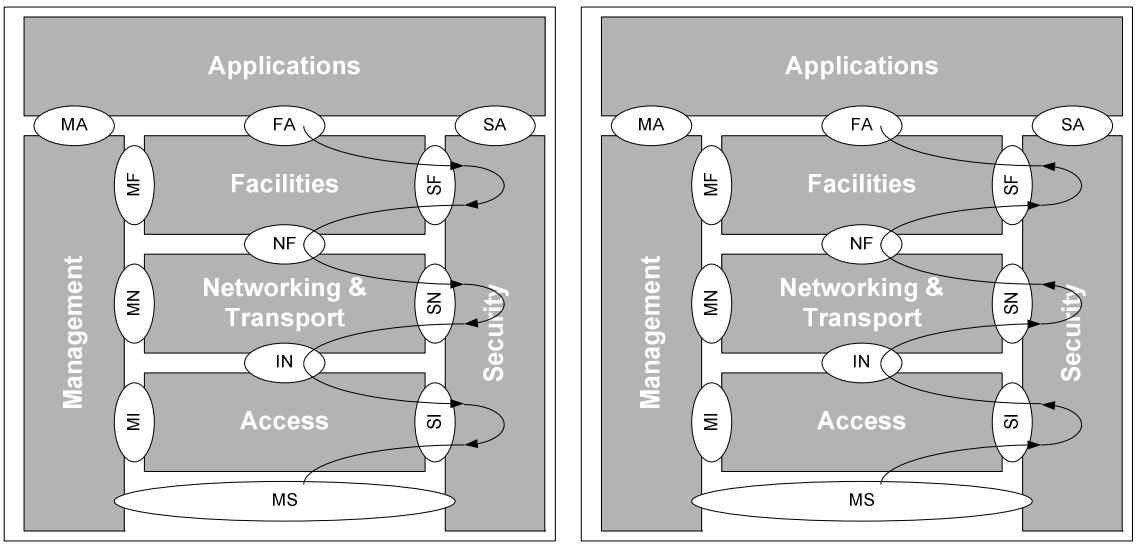

Figure 2: TX (left) and RX (right) information flow through the ITS station

The present document specifies only the SN-SAP, therefore only a subset of the ITS station reference architecture has to be taken into account. Figure 3 shows the typical information flow between any sending (TX) and receiving (RX) party, with regard to the SN-SAP only. The security entity acts like a layer inside the networking and transport layers, i.e. it is called during the processing of messages traversing the networking and transport layers. The security entity will however not act as a layer above or below the networking and transport layers. This means that interactions with Facilities and Access layers are achieved via other means, i.e. the NF-SAP is used for the interaction between the networking and transport layers and Facilities layers, whereas the IN-SAP is used for the interaction between the access layer and networking and transport layers.

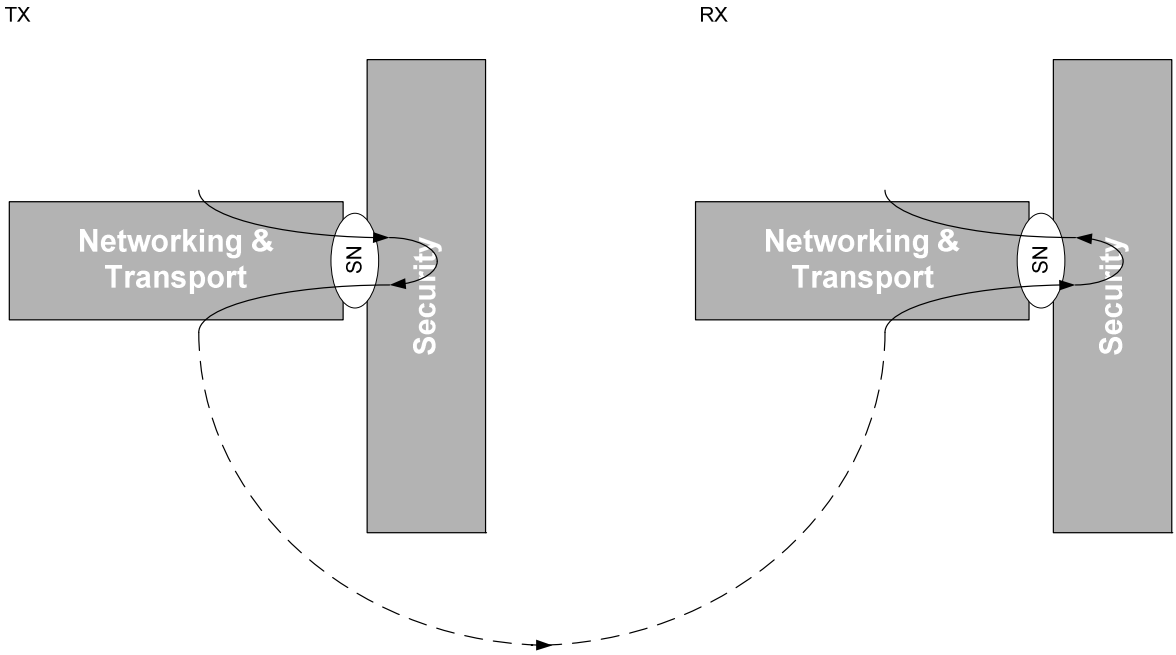

Figure 3: SN-SAP centric Information flow

4.1.3 Horizontal control communication

Figure 4 outlines the second communication principle. There is a horizontal control communication between the security entity and the corresponding communications layer, networking and transport in this case. This is needed for the ID change functionality introduced later. In general, the security entity will be able to indicate an ID change to the corresponding layer and some additional ID change related calls.

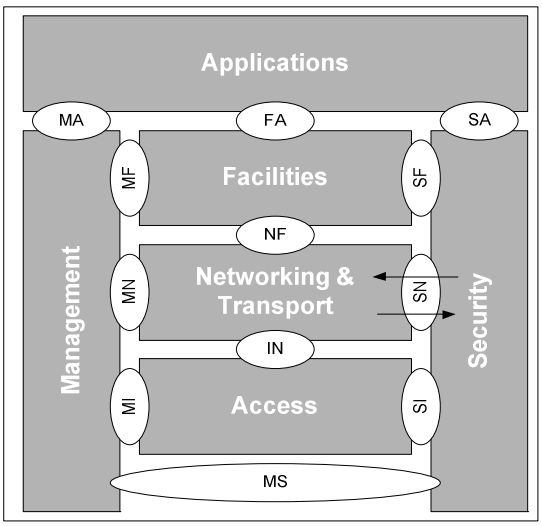

Figure 4: Horizontal Control Communication

4.1.4 Protocol work split

The SN-SAP provides a set of primitive security functions to the networking and transport layer.

Figure 5 shows how a protocol entity within the networking and transport layer handles the sending and receiving of information but uses some security extensions to invoke the primitive functions of the Security layer in order to meet the security requirements of this layer. They are supported by the Identity Management Capabilities, specified in ETSI TS 102 940 [2], clause 6, necessary to apply the Atomic Security Capabilities.

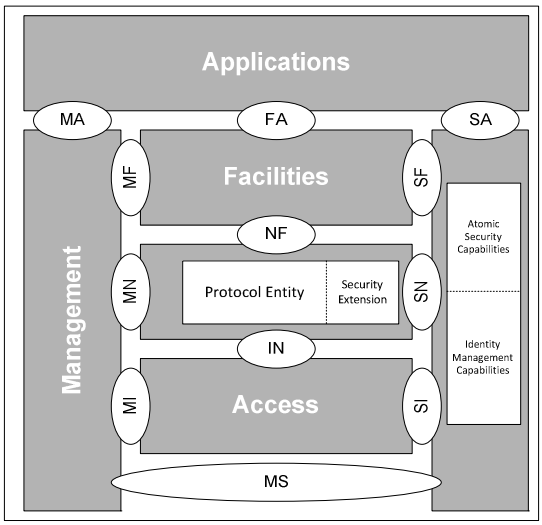

Figure 5: Protocol work split

4.1.5 Multiple instances

The present document does not discuss architecture. However, the SAP shall support different permissions. The management of different credential sets at the same time can be implemented by using multiple instances of the security entity at the same time. Different or same components in the networking and transport layers might use multiple instances of the security entity using the service primitives described in clause 5. Handling and access control of those is out of scope of the present document.

4.1.6 Error handling

The present document does not make assumptions on implementation specific error handling for using the described services. I.e. if a call of any of the described services fails for some reason, the present document does not specify if this should be handled using exceptions or any other error handling technique.

However, the present document does specify the behaviour of services that can have a positive or negative result. E.g. a SN-VERIFY can be SUCCESSFUL if the verification was successful or it can be unsuccessful, if the signature was invalid (FALSE_SIGNATURE). This is considered to be within normal operation conditions, and therefore not an error.

4.2 Security services

The required ITS security services are identified as the first level security services in ETSI TS 102 940 [2], clause 5.2. In addition to those, security services used in the research projects PRE-DRIVE C2X and EVITA where adopted and fitted to the existing services. See PRE-DRIVE C2X Deliverable D1.3 [i.4] and EVITA Deliverable D3.2 [i.5] for documentation on the research project services.

Table 1 summarizes the security services to be specified in the present document, clause 5. Those security services are invoked directly by applications or other components and layers according to ETSI TS 102 940 [2]. A "security service group" is introduced to ease the readability of the table.

Table 1: Security Service to Service Implementation Assignment

Security Service Group	Security Service Name	Type/Direction	Implemented by (clause 5)
Confidentiality	Encrypt Single Message	Request	SN-ENCRYPT
	Decrypt Single Message	Request	SN-DECRYPT
Authentication and Integrity	Authorize Single Message	Request	SN-SIGN
	Validate Authorization on Single Message	Request	SN-VERIFY
Identity Management	Lock ID Change	Request	SN-ID-LOCK
	Unlock ID Change	Request	SN-ID-UNLOCK
	Subscribe to ID Change Notification	Request	SN-IDCHANGE-SUBSCRIBE
	Unsubscribe from ID Change Notification	Request	SN-IDCHANGE-UNSUBSCRIBE
	Change ID	Indication send to subscribed entities	SN-IDCHANGE-EVENT
	Trigger ID Change	Request	SN-IDCHANGE-TRIGGER
Extras	Log Security Event	Request	SN-LOG-SECURITY-EVENT
	Extract Permissions	Request	SN-EXTRACT-PERMISSIONS
	Encapsulate Message	Request	SN-ENCAP
	Decapsulate Message	Request	SN-DECAP

5 Interfaces between the security entity and the networking and transport layers

5.1 Interface services

The following services for the SN-SAP are defined in the present document:

- SN-SIGN
  Create authentication information for outgoing ITS messages
- SN-VERIFY
  Validate authentication information from incoming ITS messages
- SN-ENCRYPT
  Encrypt outgoing ITS single messages
- SN-DECRYPT
  Decrypt incoming ITS single messages
- SN-IDCHANGE-SUBSCRIBE
  Subscribe for notifications on SN-IDCHANGE-EVENT, used for concurrent identifiers exchange across the ITS-S
- SN-IDCHANGE-EVENT
  The indication sent to subscribers on IDCHANGE
- SN-IDCHANGE-UNSUBSCRIBE
  Unsubscribe for IDCHANGE notifications, cf. SN-IDCHANGE-EVENT
- SN-IDCHANGE-TRIGGER
  Ask security entity to trigger IDCHANGE procedure
- SN-ID-LOCK
  Ask security entity to avoid IDCHANGEs
- SN-ID-UNLOCK
  Release SN-ID-LOCK

- SN-LOG-SECURITY-EVENT
  Insert external security events

- SN-ENCAP
  Encapsulate outbound messages in a security envelope. This is an alternative way of calling the same functionality that SN-SIGN and/or SN-ENCRYPT offer, where the security parameter selection is done via a security profile parameter or security entity pre-sets

- SN-DECAP
  Decapsulate inbound messages from a security envelope. This is an alternative way of calling the same functionality that SN-VERIFY and/or SN-DECRYPT offer, and should be used together with SN-ENCAP

5.2 Service primitives and parameters

5.2.1 SN-SIGN

5.2.1.1 Description

The service adds authentication information to the message. Key and identity management is internal to the security entity. Format of the created security header is dependent on the selected security protocol. The key to use is expected to be selected by the key and identity management of the security entity. Nevertheless, it is optionally possible to indicate the key to use via the key_handle parameter.

5.2.1.2 SN-SIGN.request

SN-SIGN.request is sent from the networking and transport layers to the security entity for executing the SIGN service. The parameters are described in Table 2.

Table 2: SN-SIGN.request

Name	Type	Valid range	Description	Status
tbs_message_length	INTEGER	0 to 2^{16} - 1	Length of the message to be signed	Mandatory
tbs_message	OCTET STRING	tbs_message_length octets	Octet string containing the message to be signed	Mandatory
its_aid	INTEGER	ANY	ITS-AID of the application payload or Networking & Transport management packet to determine the security profile to apply	Mandatory
permissions_length	INTEGER	0 to 2^{16} - 1	Length of the permissions	Mandatory
permissions	OCTET STRING	Maximum length of 31 octets	Specify the sender's permissions for the security entity to decide which key to use. For example, when using ETSI TS 103 097 [i.10] security protocol, the permissions contain the SSP associated with ITS-AID	Mandatory
context_information	OCTET STRING	ANY	Context information which could be used in selecting properties of the underlying security protocol for various purposes	Optional
key_handle	INTEGER	0 to 2^{64} - 1	An indicator for the security entity to decide which key to use	Optional

5.2.1.3 SN-SIGN.confirm

SN-SIGN.confirm is sent from the security entity to the networking and transport layers as a corresponding reply to SN-SIGN.request. The parameters are described in Table 3.

Table 3: SN-SIGN.confirm

Name	Type	Valid range	Description	Status
sec_message_length	INTEGER	0 to 2^{16} - 1	Length of the signed message	Mandatory
sec_message	OCTET STRING	sec_message_length octets	Octet string of the signed message	Mandatory

5.2.2 SN-VERIFY

5.2.2.1 Description

The service verifies the validity of the digital signature and meta information contained in the security header. Its format, specification, and features are dependent on the selected security protocol.

5.2.2.2 SN-VERIFY.request

SN-VERIFY.request is sent from the networking and transport layers to the security entity for executing the VERIFY service. The parameters are described in Table 4.

Table 4: SN-VERIFY.request

Name	Type	Valid range	Description	Status
sec_header_length	INTEGER	0 to 2^{16} - 1	Length of the security header	Mandatory
sec_header	OCTET STRING	sec_header_length octets	Octet string containing the security header	Mandatory
message_length	INTEGER	0 to 2^{16} - 1	Length of the message to be verified	Mandatory
message	OCTET STRING	message_length octets	Octet string containing the message to be verified	Mandatory

5.2.2.3 SN-VERIFY.confirm

SN-VERIFY.confirm is sent from the security entity to the networking and transport layers as a corresponding reply to SN-VERIFY.request. The parameters are described in Table 5.

Table 5: SN-VERIFY.confirm

Name	Type	Valid range	Description	Status
report	INTEGER	0 to 2^8 - 1	VERIFY return code: SUCCESS FALSE_SIGNATURE INVALID_CERTIFICATE REVOKED_CERTIFICATE INCONSISTENT_CHAIN INVALID_TIMESTAMP DUPLICATE_MESSAGE INVALID_MOBILITY_DATA UNSIGNED_MESSAGE SIGNER_CERTIFICATE_NOT_FOUND UNSUPPORTED_SIGNER_IDENTIFIER_TYPE INCOMPATIBLE_PROTOCOL	Mandatory
certificate_id	OCTET STRING	8 octets	Identification of the source certificate, e.g. by the certificate hash, to be forwarded to facilities	Optional
its_aid_length	INTEGER	0 to 2^{16} - 1	Length of the its_aid field	Mandatory
its_aid	INTEGER	ANY	ITS-AID of the application payload or Networking & Transport management packet to determine the security profile to apply	Mandatory
permissions	OCTET STRING	Maximum length of 31 octets	In case the used security protocol is capable of attaching senders permissions, verify may report those back to the caller. The definition is dependent on the applied security protocol. For example, when using ETSI TS 103 097 [i.10] security protocol, the permissions contain the SSP associated with ITS-AID	Mandatory

5.2.3 SN-ENCRYPT

5.2.3.1 Description

This service encrypts message for specific recipients. The designated recipient has to be known to the security entity. Therefore, an identifier is required to indicate the recipient. An internal mapping of target_id to certificate_id shall be possible, to select the proper target key.

5.2.3.2 SN-ENCRYPT.request

SN-ENCRYPT.request is sent from the networking and transport layers to the security entity for executing the ENCRYPT service. The parameters are described in Table 6.

Table 6: SN-ENCRYPT.request

Name	Type	Valid range	Description	Status
tbe_payload_length	INTEGER	0 to 2^{16} - 1	Length of the payload to be encrypted	Mandatory
tbe_payload	OCTET STRING	tbe_payload_length octets	Octet string of the Payload to be encrypted	Mandatory
target_id_list_length	INTEGER	0 to 2^{16} - 1	Length of the target_id_list	Mandatory
target_id_list	SET OF OCTET STRING	target_id_list_length elements each of 8 octets	Unordered collection of target IDs, for specifying multiple recipients	Mandatory
context_information	OCTET STRING	ANY	Context information which could be used in selecting properties of the underlying security protocol for various purposes	Optional

5.2.3.3 SN-ENCRYPT.confirm

SN-ENCRYPT.confirm is sent from the security entity to the networking and transport layers as a corresponding reply to SN- ENCRYPT.request. The parameters are described in Table 7.

Table 7: SN-ENCRYPT.confirm

Name	Type	Valid range	Description	Status
encrypted_message_length	INTEGER	0 to 2^{16} - 1	Length of the encrypted_message	Mandatory
encrypted_message	OCTET STRING	encrypted_message_length octets	Octet string of the encrypted_message	Mandatory

5.2.4 SN-DECRYPT

5.2.4.1 Description

This services decrypts messages, which were encrypted using the ENCRYPT service.

5.2.4.2 SN-DECRYPT.request

SN-DECRYPT.request is sent from the networking and transport layers to the security entity for executing the DECRYPT service. The parameters are described in Table 8.

Table 8: SN-DECRYPT.request

Name	Type	Valid range	Description	Status
encrypted_message_length	INTEGER	0 to 2^{16} - 1	Length of the encrypted_message	Mandatory
encrypted_message	OCTET STRING	encrypted_message_length octets	Octet string of the encrypted_message	Mandatory

5.2.4.3 SN-DECRYPT.confirm

SN-DECRYPT.confirm is sent from the security entity to the networking and transport layers as a corresponding reply to SN-DECRYPT.request. The parameters are described in Table 9.

Table 9: SN-DECRYPT.confirm

Name	Type	Valid range	Description	Status
plaintext_message_length	INTEGER	0 to 2^{16} - 1	Length of the decrypted message	Mandatory
plaintext_message	OCTET STRING	plaintext_message_length octets	Octet string containing the decrypted message	Mandatory
report	INTEGER	0 to 2^{8} - 1	Decrypt return code: SUCCESS UNENCRYPTED_MESSAGE DECRYPTION_ERROR INCOMPATIBLE_PROTOCOL	Mandatory

5.2.5 SN-IDCHANGE-SUBSCRIBE

5.2.5.1 Description

Subscribe for notifications on IDCHANGE, used for concurrent identifiers exchange across the ITS-S.

5.2.5.2 SN-IDCHANGE-SUBSCRIBE.request

SN-IDCHANGE-SUBSCRIBE.request is sent from the networking and transport layers to the security entity for executing the IDCHANGE-SUBSCRIBE service. The parameters are described in Table 10.

Table 10: SN-IDCHANGE-SUBSCRIBE.request

Name	Type	Valid range	Description	Status
idchange_event_hook	Not applicable in ASN.1	Not applicable in ASN.1	Callback function, which is called when an ID-change event occurs. The signature of the hook function is specified in clause 5.2.6	Mandatory
subscriber_data	OCTET STRING	ANY	Additional parameter for callback function internal use. This will be passed to the hook function on every call	Optional

5.2.5.3 SN-IDCHANGE-SUBSCRIBE.confirm

SN-IDCHANGE-SUBSCRIBE.confirm is sent from the security entity to the networking and transport layers as a corresponding reply to SN-IDCHANGE-SUBSCRIBE.request. The parameters are described in Table 11.

Table 11: SN-IDCHANGE-SUBSCRIBE.confirm

Name	Type	Valid range	Description	Status
subscription	INTEGER	0 to 2^{64} - 1	Subscription handle for unsubscribe	Mandatory

5.2.6 SN-IDCHANGE-EVENT

5.2.6.1 Description

Indication for notifications on IDCHANGE, cf. SN-IDCHANGE-SUBSCRIBE in clause 5.2.5.

5.2.6.2 SN-IDCHANGE-EVENT.indication

SN-IDCHANGE-EVENT.indication is sent from the security entity to the networking and transport layers for executing the IDCHANGE-EVENT service. The parameters are described in Table 12.

Table 12: SN-IDCHANGE-EVENT.indication

Name	Type	Valid range	Description	Status
command	ENUMERATED	PREPARE COMMIT ABORT DEREG	Id-change phase, see clause 6.3	Mandatory
id	OCTET STRING	8 octets	Id to be set	Mandatory
subscriber_data	OCTET STRING	ANY	Additional parameter for callback function internal use. This will be passed to the hook function on every call	Optional

5.2.6.3 SN-IDCHANGE-EVENT.response

SN-IDCHANGE-EVENT.response is sent from the networking and transport layers to the security entity as a corresponding reply to SN-IDCHANGE-EVENT.indication. The parameters are described in Table 13.

Table 13: SN-IDCHANGE-EVENT.response

Name	Type	Valid range	Description	Status
return_code	BOOLEAN	true or false	Acknowledgement to the given command	Mandatory

5.2.7 SN-IDCHANGE-UNSUBSCRIBE

5.2.7.1 Description

Unsubscribe for IDCHANGE notifications, cf. SN-IDCHANGE-SUBSCRIBE in clause 5.2.5.

5.2.7.2 SN-IDCHANGE-UNSUBSCRIBE.request

SN-IDCHANGE-UNSUBSCRIBE.request is sent from the networking and transport layers to the security entity for executing the IDCHANGE-UNSUBSCRIBE service. The parameters are described in Table 14.

Table 14: SN-IDCHANGE-UNSUBSCRIBE.request

Name	Type	Valid range	Description	Status
subscription	INTEGER	0 to 2^{64} - 1	Subscription handle, given through subscribe	Mandatory

5.2.7.3 SN-IDCHANGE-UNSUBSCRIBE.confirm

SN-IDCHANGE-UNSUBSCRIBE.confirm is sent from the security entity to the networking and transport layers as a corresponding reply to SN-IDCHANGE-UNSUBSCRIBE.request. The parameters are described in Table 15.

Table 15: SN-IDCHANGE-UNSUBSCRIBE.confirm

Name	Type	Valid range	Description	Status
(none)	-	-	-	-

5.2.8 SN-IDCHANGE-TRIGGER

5.2.8.1 Description

Ask security entity to trigger IDCHANGE procedure.

5.2.8.2 SN-IDCHANGE-TRIGGER.request

SN-IDCHANGE-TRIGGER.request is sent from the networking and transport layers to the security entity for executing the IDCHANGE-TRIGGER service. The parameters are described in Table 16.

Table 16: SN-IDCHANGE-TRIGGER.request

Name	Type	Valid range	Description	Status
(none)	-	-	-	-

5.2.8.3 SN-IDCHANGE-TRIGGER.confirm

SN-IDCHANGE-TRIGGER.confirm is sent from the security entity to the networking and transport layers as a corresponding reply to SN-IDCHANGE-TRIGGER.request. The parameters are described in Table 17.

Table 17: SN-IDCHANGE-TRIGGER.confirm

Name	Type	Valid range	Description	Status
(none)	-	-	-	-

5.2.9 SN-ID-LOCK

5.2.9.1 Description

Ask security entity to avoid IDCHANGEs for the number of seconds specified in duration. The lock will be released automatically afterwards or can be released by using SN-ID-UNLOCK.

5.2.9.2 SN-ID-LOCK.request

SN-ID-LOCK.request is sent from the networking and transport layers to the security entity for executing the ID-LOCK service. The parameters are described in Table 18.

Table 18: SN-ID-LOCK.request

Name	Type	Valid range	Description	Status
Duration	INTEGER	0 to 2^8 - 1	Number of seconds to lock	Mandatory

5.2.9.3 SN-ID-LOCK.confirm

SN-ID-LOCK.confirm is sent from the security entity to the networking and transport layers as a corresponding reply to SN-ID-LOCK.request. The parameters are described in Table 19.

Table 19: SN-ID-LOCK.confirm

Name	Type	Valid range	Description	Status
lock_handle	INTEGER	0 to 2^{64} - 1	Handle to unlock manually	Mandatory

5.2.10 SN-ID-UNLOCK

5.2.10.1 Description

Release SN-ID-LOCK.

5.2.10.2 SN-ID-UNLOCK.request

SN-ID-UNLOCK.request is sent from the networking and transport layers to the security entity for executing the ID-UNLOCK service. The parameters are described in Table 20.

Table 20: SN-ID-UNLOCK.request

Name	Type	Valid range	Description	Status
lock_handle	INTEGER	0 to 2^{64} - 1	Handle to unlock manually	Mandatory

5.2.10.3 SN-ID-UNLOCK.confirm

SN-ID-UNLOCK.confirm is sent from the security entity to the networking and transport layers as a corresponding reply to SN-ID-UNLOCK.request. The parameters are described in Table 21.

Table 21: SN-ID-UNLOCK.confirm

Name	Type	Valid range	Description	Status
(none)	-	-	-	-

5.2.11 SN-LOG-SECURITY-EVENT

5.2.11.1 Description

Insert external security events.

5.2.11.2 SN-LOG-SECURITY-EVENT.request

SN-LOG-SECURITY-EVENT.request is sent from the networking and transport layers to the security entity for executing the LOG-SECURITY-EVENT service. The parameters are described in Table 22.

Table 22: SN-LOG-SECURITY-EVENT.request

Name	Type	Valid range	Description	Status
event_type	ENUMERATED	see list below	Type of security event	Mandatory
neighbour_id_list_length	INTEGER	0 to 2^{32} - 1	Length of the following neighbour_id_list field	Mandatory
neighbour_id_list	SET OF OCTET STRING	neighbour_id_list_length elements each element has a length of 8 octets	List of affected V2X neighbour nodes, expressed via certificate hash	Mandatory
event_time	INTEGER	0 to 2^{32} - 1 (shall be in the past)	Timestamp of the security event	Mandatory
event_location	SEQUENCE { latitude INTEGER, longitude INTEGER }	-2^{31} to $+2^{31}$ - 1 (latitude) -2^{31} to $+2^{31}$ - 1 (longitude)	Location of the security event expressed in latitude, longitude	Optional
event_evidence_list_length	INTEGER	0 to 2^{32} - 1	Length of the following event_evidence_list field	Optional
event_evidence_list	SET OF { length and OCTET STRING}	ANY	Signed CAMs or DENMs can be used to proof the existence of the neighbour node at stated time and position. This information can be used to prevent blackmailing attacks by malicious applications	Optional
event_evidence_type	ENUMERATED	CAM, DENM, etc.	Type of the attached event_evidence_content	Optional
event_evidence_content_length	INTEGER	0 to 2^{32} - 1	Length of the following event_evidence_content field	Optional
event_evidence_content	OCTET STRING	event_evidence_content_ length octets	Attached evidence for the event	Optional

The following event_type elements consider security related data verifications on receiver side that may be used to report detected misbehaviour.

- TIME_CONSISTENCY_FAILED: consistency check of timestamps contained in different parts of a packet failed. This may occur if an attacker manipulates the timestamps on one layer of a sender station and the receiver detects the inconsistency with redundant information. For example, a check on receiver's application layer detects that the generation time of a single-hop message differs a lot from the generation time that was added to lower layers (e.g. network header, security header).

- LOCATION_CONSISTENCY_FAILED: consistency check of location data contained in different parts of a packet failed. This may occur if an attacker manipulates the location on one layer of a sender station and the receiver detects the inconsistency with redundant information. For example, a check on receiver's application layer detects that the location of a single-hop message differs a lot from the location that was added to lower layers (e.g. network header, security header).

- ID_CONSISTENCY_FAILED: consistency check of identifiers contained in different parts of a packet failed. This may occur if an attacker manipulates the identifier on one layer of a sender station and the receiver detects the inconsistency with redundant information. For example, a check on receiver's application layer detects that the identifier of a single-hop message differs from the identifier that was added to lower layers (e.g. network header, security header).

- DISALLOWED_MESSAGE_CONTENT: a message-based plausibility check is using predefined rules and physical boundaries. These checks are using a transmitted location data that contains the position of the sender, its current speed and heading at a specific point in time.

- In basic checks the values of given location data are compared with the predefined domain of definition. The heading value shall follow the domain of definition according to related standardization for CAM and DENM as well as for network layer headers. A heading value larger than 360° for example should be considered to be not plausible. Furthermore, the velocity values shall be checked as well as the WGS84 encoded latitude and longitude value of a sender's position. For example, the velocity of a vehicle less than -30 m/s and greater than 100 m/s is suspicious in normal road traffic.

- DISALLOWED_MESSAGE_FREQUENCY: a plausibility check on the receiving station is able to count the received messages from the direct neighbours and is able to detect violations according to ETSI TS 102 637-1 [i.6].

- REPLAY_DETECTION_TIME: in a time- replay check, the maximum transmission delay shall be verified at the receiving station. According to ETSI TS 102 637-2 [i.7], the maximum transmission delay of CAMs shall not be greater than 100 ms. As a result, messages with an outdated timestamp or a future timestamp can be seen as not plausible. The check aims to detect time-based replay attacks where an attacker records a valid message at time T1 and replays it later at time T2.

- REPLAY_DETECTION_LOCATION: in a communication range check, the distance between a single-hop sender and the own position of the receiver is calculated. If this distance is greater than the maximum transmission range of a radio that is following the maximum specified transmission power according to ETSI ES 202 663 [i.8], the location of the sender can be assumed to be not plausible. This kind of check aims to detect location-based replay attacks that are also known as tunnel or wormhole attack. Here, an attacker records a valid message at location L1, transmits the message quickly to location L2 and re-broadcasts it there.

- MOVEMENT_PLAUSIBILITY: based on a physical mobility model for vehicles a position can be predicted using previously received position statements. When a new message is received, the predicted position can be compared with the claimed position whereupon large deviations are suspicious and may result in misbehaviour detection. As CAMs are broadcasted with a maximum frequency of 10 Hz, an accurate position vector of the next CAM can be assumed. By checking the movement plausibility, position jumps and unexpected mobility behaviour can be detected.

- APPEARANCE_PLAUSIBILITY: in normal traffic conditions, it can be assumed that new vehicles first appear at the boundary of the communication range. As a result, a first single-hop from a station with an unknown ID shall contain a location data that states a certain distance between the sender's station and the own receiver station. However, pseudonym changes and hidden stations, caused possibly by large buildings in urban traffic, require a context depended check of sudden appearing stations.

- LOCATION_PLAUSIBILITY_SENSOR: if a received position of a neighbour node can be mapped to an object detected by a local sensor, then this vehicle position can be assumed to be trustworthy. On the other hand, the object detection of a local environment sensor can be used to dispute a claimed location. If a neighbour vehicle claims a position that is located between the own station and an object that is detected by the radar, then this vehicle position is not trustworthy.

- LOCATION_PLAUSIBILITY_MAP: a digital road map can be used to check the position of a sending vehicle station assuming every receiving ITS station is equipped with a map. However, a vehicle that cannot be assigned to a valid road segment of the local map is possibly driving on a private road or is parked beside a road. It has to be further considered that the local map may be outdated.

- LOCATION_PLAUSIBILITY_CONTRADICTION: a station that receives contradictory information from two different, but equally trusted nodes cannot directly determine which statement is true and which is false. However, by collecting additional information about the same or a similar statement from different independent senders, the receiver may be able to take a decision assuming that the majority of provided information is correct:

    - LOCATION_PLAUSIBILITY_CONTRADICTION_VEHICLE_DIMENSION: as vehicles are regularly broadcasting CAMs with their absolute position and their rough stations' dimensions, a check of position overlaps can be performed by comparing the location data of nearby stations.

    - LOCATION_PLAUSIBILITY_CONTRADICTION _NEIGHBOR_INFO: neighbours may distribute their local first-hand information (e.g. radar-tracked nodes) or reputation information about their neighbour nodes. A receiver of this information is able to compare the received tables with other received tables and with its local neighbour information.

Applications may specify additional types that are related to specific implausibilities, e.g. detection of attackers sending contradicting event notifications.

The interface may further consider security events on sender side that could lead to a deactivation of the own security subsystem.

5.2.11.3 SN-LOG-SECURITY-EVENT.confirm

SN-LOG-SECURITY-EVENT.confirm is sent from the security entity to the networking and transport layers as a corresponding reply to SN-LOG-SECURITY-EVENT.request. The parameters are described in Table 23.

Table 23: SN-LOG-SECURITY-EVENT.confirm

Name	Type	Valid range	Description	Status
(none)	-	-	-	-

5.2.12 SN-ENCAP

5.2.12.1 Description

Encapsulate outbound messages in a security envelope. This is an alternative way of calling the same functionality that SN-SIGN and/or SN-ENCRYPT offer, where the security parameter selection is done via a security profile parameter or security entity pre-sets.

5.2.12.2 SN-ENCAP.request

The service primitive SN-ENCAP.request is sent from the networking and transport layers to the security entity for executing the ENCAP service. The parameters are described in Table 24.

Table 24: SN-ENCAP.request

Name	Type	Valid range	Description	Status
tbe_packet_length	INTEGER	0 to 2^{16} - 1	Length of the packet to encapsulate into the security envelop	Mandatory
tbe_packet	OCTET STRING	tbe_packet_length octets	The packet to be encapsulated into the security envelop	Mandatory
sec_services	INTEGER	0 to 2^{16} - 1	The security service(s) to invoke	Optional
its_aid _length	INTEGER	0 to 2^{16} - 1	Length of the its_aid field	Optional
its_aid	INTEGER	ANY	ITS-AID of the application payload or Networking & Transport management packet to determine the security profile to apply	Mandatory
permissions	OCTET STRING	Maximum length of 31 octets	Specify the senders permissions for the security entity to decide which key to use. For example, when using ETSI TS 103 097 [i.10] security protocol, the permissions contain the SSP associated with ITS-AID	Mandatory
context_information	OCTET STRING	ANY	Context information which could be used in selecting properties of the underlying security protocol for various purposes	Optional
target_id_list_length	INTEGER	0 to 2^{16} - 1	Length of the target_id_list	Optional
target_id_list	SET OF OCTET STRING	target_id_list_length elements each of 8 octets	Unordered collection of target IDs, for specifying multiple recipients	Optional

5.2.12.3 SN-ENCAP.confirm

The service primitive SN-ENCAP.confirm is sent from the security entity to the networking and transport layers as a corresponding reply to a SN-ENCAP.request. The parameters are described in Table 25.

Table 25: SN-ENCAP.confirm

Name	Type	Valid range	Description	Status
sec_packet_length	INTEGER	0 to 2^{16} - 1	Length of the Secured Packet	Mandatory
sec_packet	OCTET STRING	sec_packet_length octets	The Secured Packet	Mandatory

5.2.13 SN-DECAP

5.2.13.1 Description

Decapsulate inbound messages from a security envelope. This is an alternative way of calling the same functionality that SN-VERIFY and/or SN-DECRYPT offer, and should be used together with SN-ENCAP.

5.2.13.2 SN-DECAP.request

The service primitive SN-DECAP.request is sent from the networking and transport layers to the security entity for executing the DECAP service. The parameters are described in Table 26.

Table 26: SN-DECAP.request

Name	Type	Valid range	Description	Status
sec_packet_length	INTEGER	0 to 2^{16} - 1	Length of the Secured Packet	Mandatory
sec_packet	OCTET STRING	sec_packet_length octets	Octet string containing the Secured Packet	Mandatory

5.2.13.3 SN-DECAP.confirm

The service primitive SN-DECAP.confirm is sent from the security entity to the networking and transport layers as a corresponding reply to a SN-DECAP.request. The parameters are described in Table 27.

Table 27: SN-DECAP.confirm

Name	Type	Valid range	Description	Status
plaintext_packet_length	INTEGER	0 to 2^{16} - 1	Length of the decrypted and verified packet	Mandatory
plaintext_packet	OCTET STRING	plaintext_packet_length octets	The decrypted and verified packet	Mandatory
report	INTEGER	0 to 2^{8} - 1	Verify and decrypt return code: SUCCESS FALSE_SIGNATURE INVALID_CERTIFICATE REVOKED_CERTIFICATE INCONSISTENT_CHAIN INVALID_TIMESTAMP DUPLICATE_MESSAGE INVALID_MOBILITY_DATA UNSIGNED_MESSAGE SIGNER_CERTIFICATE_NOT_FOUND UNSUPPORTED_SIGNER_IDENTIFIER_TYPE INCOMPATIBLE_PROTOCOL UNENCRYPTED_MESSAGE DECRYPTION_ERROR INCOMPATIBLE_PROTOCOL	Mandatory
certificate_id	OCTET STRING	8 octets	Identification of the source certificate, e.g. by the certificate hash, to be forwarded to facilities	Optional
its_aid_length	INTEGER	0 to 2^{16} - 1	Length of the its_aid field	Mandatory
its_aid	INTEGER	ANY	ITS-AID of the application payload or Networking & Transport management packet to determine the security profile to apply	Mandatory

Name	Type	Valid range	Description	Status
permissions	OCTET STRING	Maximum length of 31 octets	In case the used security protocol is capable of attaching the senders permissions, the DECAP service may report those back to the caller. For example, when using ETSI TS 103 097 [i.10] security protocol, the permissions contain the SSP associated with ITS-AID	Mandatory

6 SN-SAP procedures

6.1 Outbound message handling

6.1.1 Using SN-SIGN and SN-ENCRYPT

This clause specifies which service primitives can be used to secure outbound communication. Two different models can be distinct. First, using SN-SIGN and SN-ENCRYPT and second using SN-ENCAP, see clause 6.1.2.

The N&T Layer Implementation can choose to add authorization information, using SN-SIGN (see Figure 6), and to encrypt a message, using SN-ENCRYPT (see Figure 7). The decision for one or the other can be taken e.g. based on the transmission mode, unicast, or broadcast, because broadcast ITS communications is unencrypted by default.

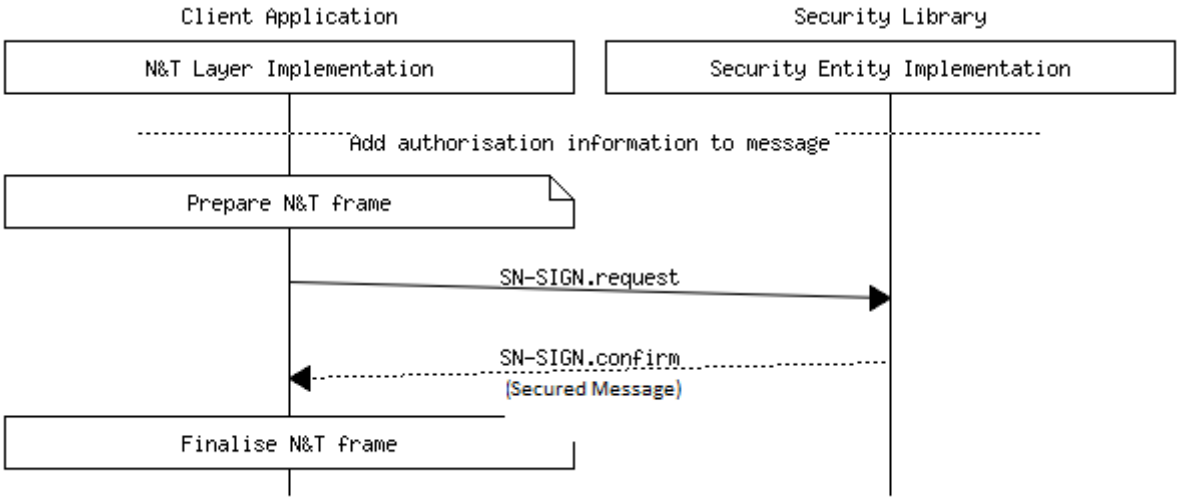

Figure 6: Using SN-SIGN

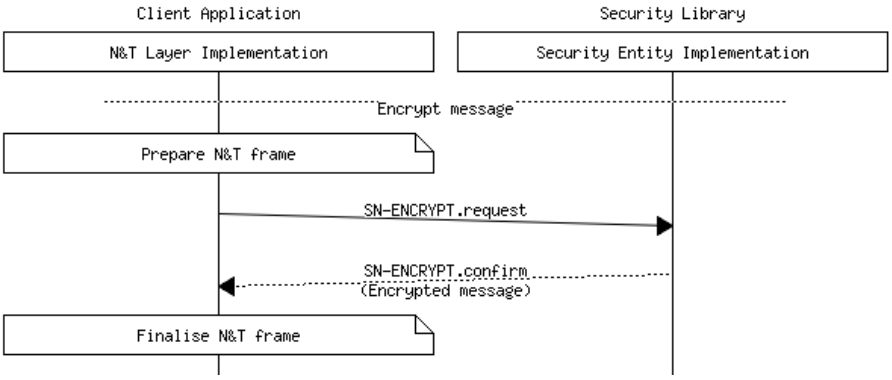

Figure 7: Using SN-ENCRYPT

6.1.2 Using SN-ENCAP

The difference in using SN-ENCAP over SN-SIGN and/or SN-ENCRYPT is that the selection of adding authorization information or encrypting a message is done inside the Security Entity Implementation. Therefore, SN-ENCAP always returns a security envelope, instead of a security header or encrypted message.

6.2 Inbound message handling

6.2.1 Using SN-VERIFY and SN-DECRYPT

This clause specifies which service primitives can be used to secure inbound communication. Two different models can be distinct. First, using SN-VERIFY and SN-DECRYPT and second using SN-DECAP, see clause 6.2.2.

When a N&T frame is received, the N&T Layer Implementation can verify the sender authentication information by using the SN-VERIFY service (see Figure 8) or decrypt encrypted messages using the SN-DECRYPT service (see Figure 9). The distinction can be made e.g. based on the transmission mode, unicast, or broadcast, because broadcast ITS communications is unencrypted by default.

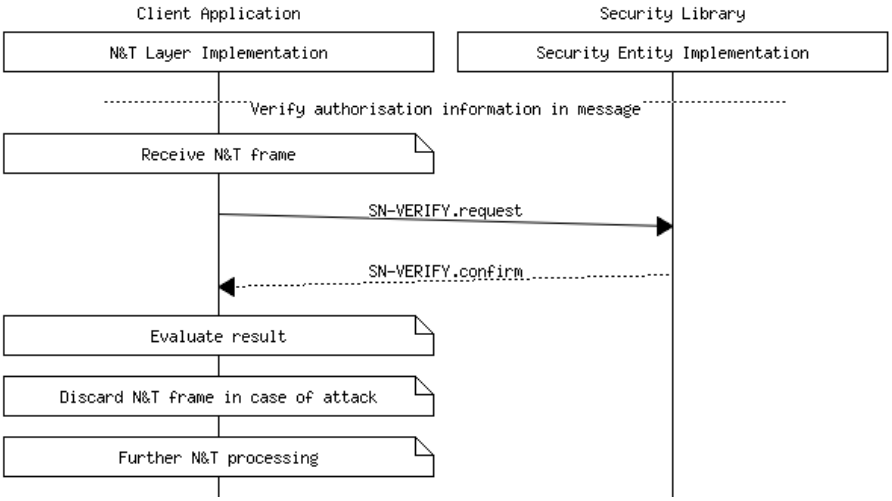

Figure 8: Using SN-VERIFY

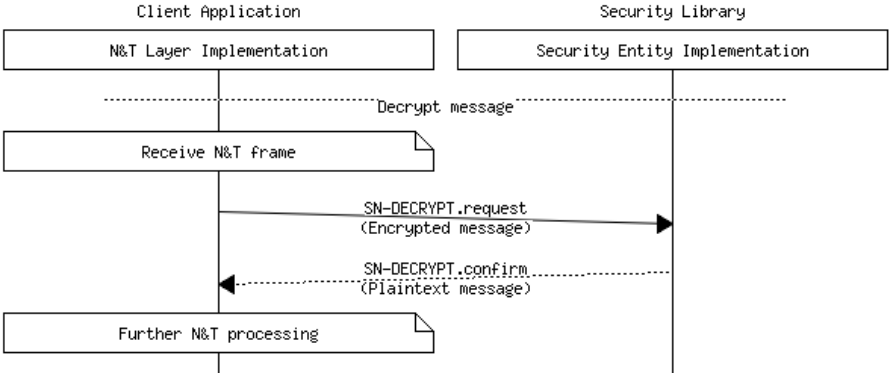

Figure 9: Using SN-DECRYPT

6.2.2 Using SN-DECAP

The SN-DECAP service corresponds to the SN-ENCAP service on the outbound side and shall decapsulate the security envelope and verify and/or decrypt its contents.

6.3 ID Management

6.3.1 IDCHANGE Notifications

6.3.1.1 Introduction

Changing authorization tickets in the communication stack may only provide unlinkable pseudonymity, if all identifiers are exchanged at the same time. Therefore, all the IDs associated with a node across different layers of the ITS stack shall be changed synchronously using the IDCHANGE Notification procedure. All layers and components, which use an ID, shall register for the IDCHANGE notifications, using the SN-IDCHANGE-SUBSCRIBE service. When the security entity indicates an identifier change event, all registered layers, and components shall invoke a two-phase commit process.

6.3.1.2 Id-change event hook

Each component, which wants to be notified by ID changes, has to offer a callback "Hook Function". This function shall accept the following commands:

a) PREPARE
Prepare for upcoming IDCHANGE

b) COMMIT
Commit IDCHANGE now

c) ABORT
IDCHANGE is aborted

d) DEREG
Registration cancelled by Security Entity

6.3.1.3 Two phase commit process

1) Subscription
For the N&T Layer subscription see Figure 10.

2) Two phase notification
Outlined in Figure 11, Figure 12 and Figure 13.
An ID change notification is done in a two-phase commit process. First, the Hook Function is called with a PREPARE command. When all registered hooks have successfully responded, i.e. returned the corresponding hook function, the Hook Function is called a second time, with the COMMIT command. Abort can occur for different reasons illustrated in Figure 12 and Figure 13.

To avoid race conditions, sending of messages with old identifiers between PREPARE and COMMIT shall be avoided and caches shall be flushed.

3) Unsubscribe
Outlined in Figure 14.
If a component wants to unsubscribe, it may do so by using the SN-IDCHANGE-UNSUBSCRIBE service.

4) Deregistration
Outlined in Figure 15.
Deregistration may also be invoked by the security entity.

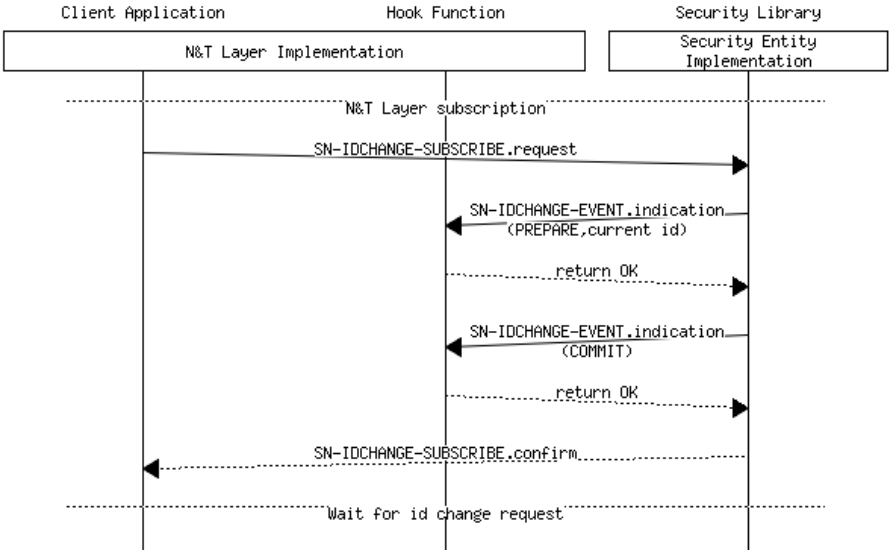

Figure 10: Subscription

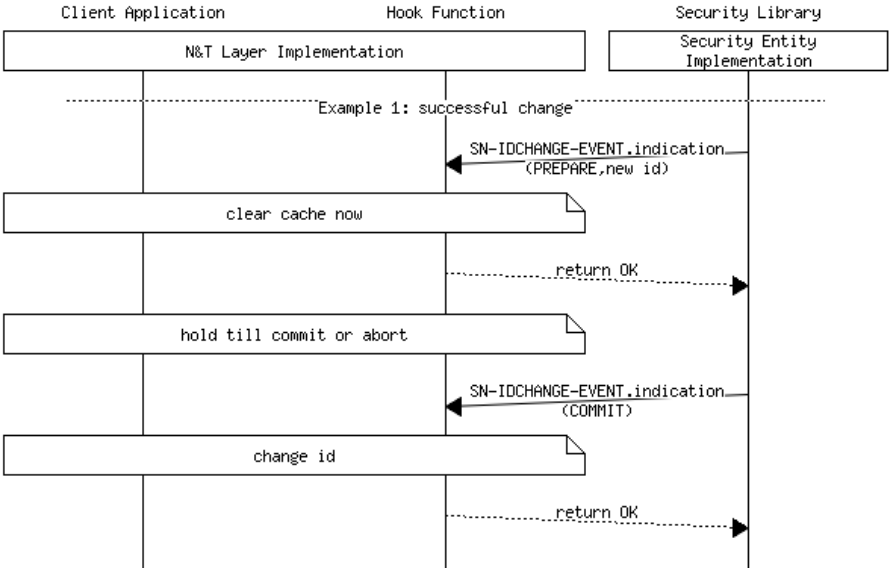

Figure 11: Notification and successful change

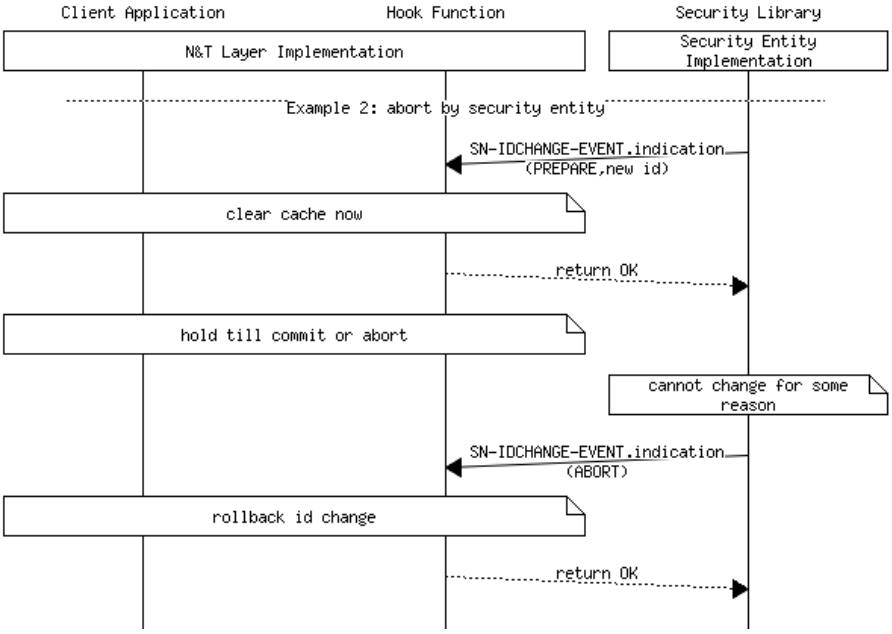

Figure 12: Notification and abort by security entity

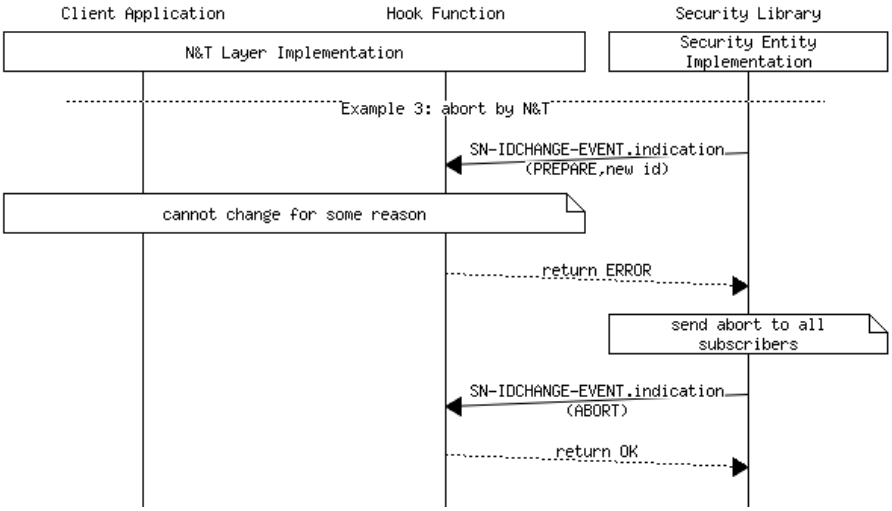

Figure 13: Notification and abort by N&T

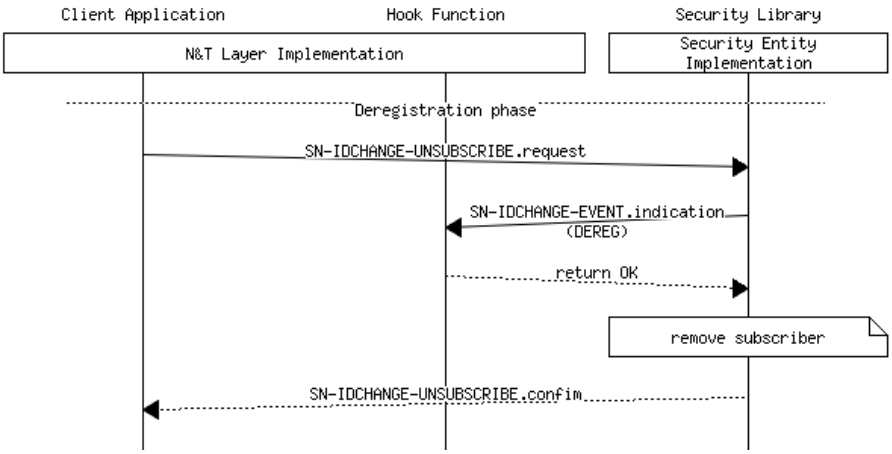

Figure 14: Unsubscribe by networking and transport layers

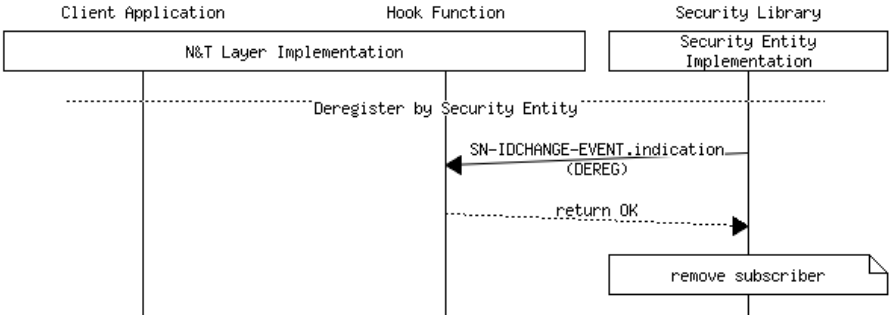

Figure 15: Deregistration by Security Entity

6.3.2 Prevent IDCHANGES

Call ID-LOCK to prevent the security entity from invoking IDCHANGES. Call ID-UNLOCK to unlock. The flow is shown in Figure 16.

NOTE 1: This could be used for example during sending of DENMs. They include a fixed action ID derived from the node ID.

NOTE 2: Safety applications such as collision avoidance applications ICRW and LCRW (ETSI TS 101 539-2 [i.2], ETSI TS 101 539-3 [i.3]) can utilize the inhibition of the pseudonym identities change, when the vehicle detects another vehicle in the safety area and the application enters the Watch state or Assist state.

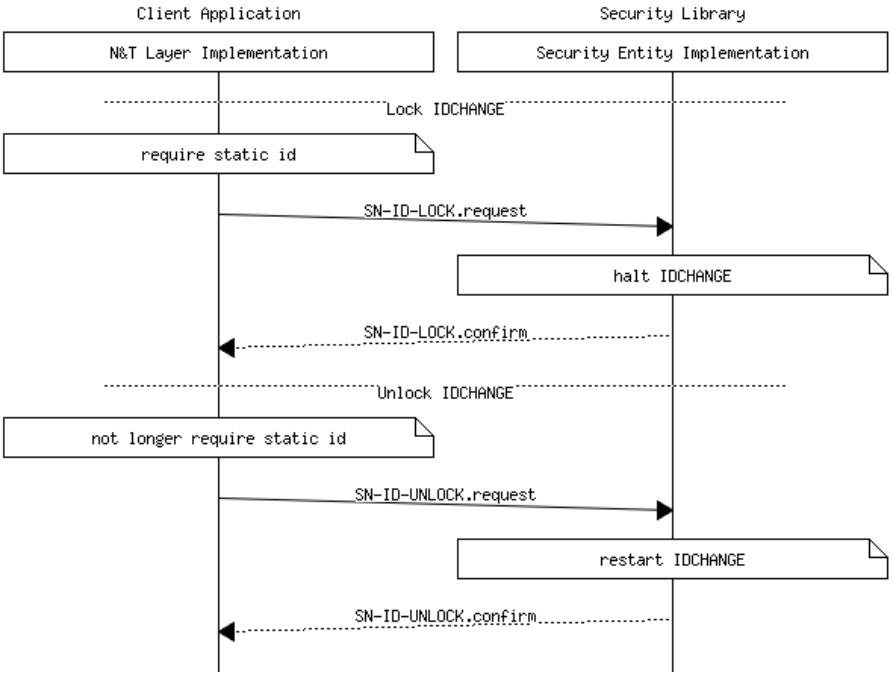

Figure 16: Using ID-LOCK and ID-UNLOCK

6.3.3 Trigger IDCHANGES

Call IDCHANGE-TRIGGER to trigger the security entity to invoke an IDCHANGE. The flow is shown in Figure 17.

NOTE: This will not lead to an immediate IDCHANGE. The IDCHANGE two phase commit above will be invoked.

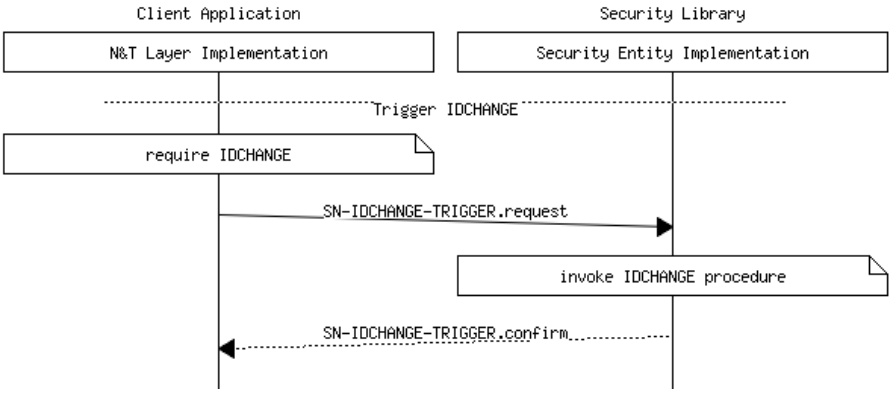

Figure 17: Using IDCHANGE-TRIGGER

6.4 Log security event

The security layer shall provide an interface that enables a stack layer to send a notification about a detected security event by the layer.

Validation of plausibility of commonly used data (i.e. mobility and location information) is part of the Secure Entity. Nevertheless, additional checks related to specific applications cannot be applied in the security stack due to missing application context information as well as data from higher layers.

EXAMPLE 1: Logging of routing attacks by the networking and transport layers.

EXAMPLE 2: Logging of attacks on transport protocols by the networking and transport layers.

EXAMPLE 3: Logging of inconsistencies in received messages by the facilities layers (e.g. compare sender's location provided on network layer with sender's location provided on facilities layer).

EXAMPLE 4: Logging of inconsistencies in application specific data related to the applications context.

The plausibility validation service of the Security Entity can subsequently use the provided security event information to mount appropriate countermeasures.

Annex A (informative): SN-Command

A.1 Overview

Table A.1 provides the relation between SN-Command.No (SN-Command reference number) and security service.

This annex provides an illustration of service primitives description using the framework of ISO 24102-3 [i.9]. This annex gives an example of ASN.1 code for SN-Command.

Table A.1: SN-Command.No

SN-Command.No	SN-Command Name	Description
0	SN-IDCHANGE-EVENT	Change ID
1 to 224		Reserved for future use.
225 to 255		For private non-standardized use.

A.2 Description

A.2.1 SN-IDCHANGE-EVENT service: SN-COMMAND.request (see clause 5.2.6.2)

Table A.2: SN-COMMAND.request

Name	Type	Valid range	Description	Status
CommandRef	INTEGER	0 to 2^{16} - 1	The cyclic number used as identifier of the SN-REQUEST.request, also used in the corresponding SN-REQUEST.confirm	Mandatory
SN-Command.No	INTEGER	0 to 255	Reference number of the security service SN-IDCHANGE-EVENT	Mandatory
command	OCTET STRING	PREPARE COMMIT ABORT DEREG	Id-change phase, see clause 6.3	Mandatory
id	OCTET STRING	8 octets	Id to be set	Mandatory
subscriber_data	OCTET STRING	ANY	Additional parameter for callback function internal use. This will be passed to the hook function on every call	Optional

NOTE: The parameters command, id and subscriber-data above are the specific function in the SN-COMMAND.request identified by the registered value of SN-Command.No of this service.

A.2.2 SN-IDCHANGE-EVENT service: SN-COMMAND.confirm (see clause 5.2.6.3)

Table A.3: SN-COMMAND.confirm

Name	Type	Valid range	Description	Status
CommandRef	INTEGER	0 to 2^{16} - 1	The cyclic number used as identifier of the SN-REQUEST.request, also used in the corresponding SN-REQUEST.confirm	Mandatory
SN-Command.No	INTEGER	0 to 255	Reference number of the security service SN-IDCHANGE-EVENT	Mandatory
return_code	ErrStatus	0 to 255 0: success 1: unspecified failure	Acknowledgement to the given command	Mandatory

Annex B (informative): SN-Request

B.1 Overview

Table B.1 provides the relation between SN-Request.No (SN-Request reference number) and data accessed.

This annex provides an illustration of service primitives description using the framework of ISO 24102-3 [i.9]. This annex gives an example of ASN.1 code for Request.

Table B.1: SN-Request.No

SN-Request.No	SN-Request Name	Description
0	SN-ENCRYPT	SendEncrypted Data
1 to 224		Reserved for future use
225 to 255		For private non-standardized use

B.2 Description

B.2.1 SN-ENCRYPT service: SN-REQUEST.request (see clause 5.2.3.2)

Table B.2: SN-REQUEST.request

Name	Type	Valid range	Description	Status
CommandRef	INTEGER	0 to 2^{16} - 1	The cyclic number used as identifier of the SN-REQUEST.request, also used in the corresponding SN-REQUEST.confirm	Mandatory
SN-Request.No	ENUMERATED	To be fixed later in registration table	Reference number of the security service SN-ENCRYPT	Mandatory
tbe_payload_length	INTEGER	0 to 2^{16} - 1	Length of the payload to be encrypted	Mandatory
tbe_payload	OCTET STRING	tbe_payload_length octets	Octet string of the Payload to be encrypted	Mandatory
target_id_list_length	INTEGER	0 to 2^{16} - 1	Length of the target_id_list	Mandatory
target_id_list	SET OF OCTET STRING	target_id_list_length elements each of 8 octets	Unordered collection of target IDs, for specifying multiple recipients	Mandatory
context_information	OCTET STRING	ANY	Context information which could be used in selecting properties of the underlying security protocol for various purposes	Optional

B.2.2 SN-ENCRYPT service: SN-REQUEST.confirm (see clause 5.2.3.3)

Table B.3: SN-REQUEST.confirm

Name	Type	Valid range	Description	Status
CommandRef	INTEGER	0 to 2^{16} - 1	The cyclic number used as identifier of the SN-REQUEST.request, also used in the corresponding SN-REQUEST.confirm	Mandatory
SN-Request.No	ENUMERATED	To be fixed later in registration table	Reference number of the security service SN-ENCRYPT	Mandatory
encrypted_message_length	INTEGER	0 to 2^{16} - 1	Length of the encrypted_message	Mandatory
encrypted_message	OCTET STRING	encrypted_message_length octets	Octet string of the encrypted_message	Mandatory
return_code	ErrStatus	0 to 255 0: success 1: unspecified failure	Acknowledgement to the given command	Mandatory

Annex C (informative): Example of service primitives description in the framework of ISO 24102-3

C.1 Overview

C.1.1 Introduction

Below is an implementation of the above service primitive functions in the framework of ISO 24102-3 [i.9].

ETSI is fully responsible for the functions (ASN.1 type definitions), i.e.:

```
SN-COMMAND.request(function(RefSNSAP-C, ...))
SN-COMMAND.confirm(function(RefSNSAP-C, ...))
SN-REQUEST.request(function(RefSNSAP-R, ...))
SN-REQUEST.confirm(function(RefSNSAP-R, ...))
```

Every function is identified by a unique reference number. Reference numbers are assigned by ISO, and published on the ISO web http://standards.iso.org/iso/24102/-3.

C.1.2 Class for SN-SAP Command.request service primitive functions

```
-- Class for SN-SAP Command.request service primitive functions
SNSAP-CR::=CLASS {
      &mxref RefSNSAP-C UNIQUE,
      &MXParam
      }
-- Named INTEGER constants identify uniquely the functions of the COMMAND service
RefSNSAP-C::=INTEGER {
    c-SN-C-IDCHANGE-EVENT   (0)
    } (0..255)
-- The generic COMMAND.request service primitive
SN-Command-request::=SEQUENCE{
    commandRef  CommandRef, -- see ISO 24102-3 (not related to a specific function)
    ref     SNSAP-CR.&mxref({SN-Command}),
    command-param   SNSAP-CR.&MXParam({SN-Command}{@ref})
    }
-- Extendible list of available functions; no need to list all functions in an implementation; only
those, which are needed and used. ",..." is the extension sign.
SN-Command  SNSAP-CR::={sn-IDCHANGE-EVENT-req, ...}
sn-IDCHANGE-EVENT-req SNSAP-CR::={&mxref c-SN-C-IDCHANGE-EVENT, &MXParam SN-idchange-event-req}
-- here we can add further functions

-- Definition of a specific function SN-idchange-event-req identified by the reference number c-SN-
C-IDCHANGE-EVENT
SN-idchange-event-req::=SEQUENCE{
    id OCTET STRING (SIZE(8)),
    subscriber_data OCTET STRING
}
```

C.1.3 Class for SN-SAP Command.confirm service primitive functions

```
-- Class for SN-SAP Command.confirm service primitive functions
-- SNSAP-CC::=CLASS {
      &mxref RefSNSAP-C UNIQUE, -- using the same named INTEGER constants as reference
      &MXParam
      }
-- The generic confirm service primitive
```

```
SN-Command-confirm::=SEQUENCE{
    commandRef      CommandRef, -- see ISO 24102-3  (not related to a specific function)
    ref         SNSAP-CC.&mxref({SN-CmdConfirm}),
    cmdConfirm-param    SNSAP-CC.&MXParam({SN-CmdConfirm}{@ref}),
    errStatus       ErrStatus -- see ISO 24102-3  (not related to a specific function)
    }
-- Extendible list of available functions
-- SN-CmdConfirm SNSAP-CC::={sn-IDCHANGE-EVENT-cnf, ...}
sn-IDCHANGE-EVENT-cnf SNSAP-CC::={&mxref c-SN-C-IDCHANGE-EVENT, &MXParam SN-idchange-event-cnf}
-- here we can add further functions
SN-idchange-event-cnf::=SEQUENCE{

    }
```

C.1.4 Class for SN-SAP Request.request service primitive functions

```
-- SN-SAP Request.request --
-- SNSAP-RR::=CLASS {
      &mxref RefSNSAP-R UNIQUE,
      &MXParam
      }
-- Named INTEGER constants identify uniquely the functions of the REQUEST service
RefSNSAP-R::=INTEGER {
    c-SN-R-ENCRYPT  (0)
    } (0..255)

SN-Request-request::=SEQUENCE{
    commandRef  CommandRef,
    ref     SNSAP-RR.&mxref({SN-Request}),
    request-param   SNSAP-RR.&MXParam({SN-Request}{@ref})
    }
SN-Request SNSAP-RR::={sn-ENCRYPT-req, ...}
sn-ENCRYPT-req SNSAP-RR::={&mxref c-SN-R-ENCRYPT, &MXParam SN-encrypt-req}
SN-encrypt-req::=SEQUENCE{
    an-Request.No   INTEGER(0..65535),
    tbe_payload OCTET STRING (SIZE(0..65535)),
    target_id       OCTET STRING (SIZE(8)),
    target_id_list  SET OF OCTET STRING OPTIONAL,
    context_information OCTET STRING OPTIONAL
    }
```

C.1.5 Class for SN-SAP Request.confirm service primitive functions

```
-- SN-SAP Request.confirm --
-- SNSAP-RC::=CLASS {
      &mxref RefSNSAP-R UNIQUE,
      &MXParam
      }
-- SN-Request-confirm::=SEQUENCE{
    commandRef      CommandRef,
    ref         SNSAP-RC.&mxref({SN-ReqConfirm}),
    reqConfirm-param    SNSAP-RC.&MXParam({SN-ReqConfirm}{@ref}),
    errStatus       ErrStatus
    }
-- SN-ReqConfirm SNSAP-RC::={sn-ENCRYPT-cnf, ...}
sn-ENCRYPT-cnf SNSAP-RR::={&mxref c-SN-R-ENCRYPT, &MXParam SN-encrypt-cnf}
SN-encrypt-cnf::=SEQUENCE{
    an-Request.No   INTEGER(0..65535),
    encrypted_message   OCTET STRING (SIZE(0..65535))
    }
```

History

Document history		
V1.1.1	April 2016	Publication