ETSI TS 102 723-9 V1.1.1 (2021-03)

TECHNICAL SPECIFICATION

Intelligent Transport Systems (ITS);
OSI cross-layer topics;
Part 9: Interface between security entity and facilities layer

Reference

DTS/ITS-00553

Keywords

adaption, addressing, interface, ITS, security

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

Important notice

Copyright Notification

Contents

Intellectual Property Rights 6

Foreword 6

Modal verbs terminology 6

Introduction 6

1 Scope 8

2 References 8
2.1 Normative references 8
2.2 Informative references 8

3 Definition of terms, symbols and abbreviations 9
3.1 Terms 9
3.2 Symbols 9
3.3 Abbreviations 9

4 Architecture 10
4.1 General 10
4.1.1 Introduction 10
4.1.2 Vertical message flow 10
4.1.3 Horizontal control communication 11
4.1.4 Protocol work split 12
4.1.5 Multiple instances 12
4.1.6 Error handling 12
4.2 Security services 12

5 Interfaces between the Security entity and the Facilities layer 13
5.1 Interface services 13
5.2 Service primitives and parameters 14
5.2.1 SF-SIGN 14
5.2.1.1 Description 14
5.2.1.2 SF-SIGN.request 14
5.2.1.3 SF-SIGN.confirm 15
5.2.2 SF-VERIFY 15
5.2.2.1 Description 15
5.2.2.2 SF-VERIFY.request 15
5.2.2.3 SF-VERIFY.confirm 15
5.2.3 SF-ENCRYPT 16
5.2.3.1 Description 16
5.2.3.2 SF-ENCRYPT.request 16
5.2.3.3 SF-ENCRYPT.confirm 17
5.2.4 SF-DECRYPT 17
5.2.4.1 Description 17
5.2.4.2 SF-DECRYPT.request 17
5.2.4.3 SF-DECRYPT.confirm 17
5.2.5 SF-IDCHANGE-SUBSCRIBE 17
5.2.5.1 Description 17
5.2.5.2 SF-IDCHANGE-SUBSCRIBE.request 18
5.2.5.3 SF-IDCHANGE-SUBSCRIBE.confirm 18
5.2.6 SF-IDCHANGE-EVENT 18
5.2.6.1 Description 18
5.2.6.2 SF-IDCHANGE-EVENT.indication 18
5.2.6.3 SF-IDCHANGE-EVENT.response 18
5.2.7 SF-IDCHANGE-UNSUBSCRIBE 19
5.2.7.1 Description 19
5.2.7.2 SF-IDCHANGE-UNSUBSCRIBE.request 19
5.2.7.3 SF-IDCHANGE-UNSUBSCRIBE.confirm 19
5.2.8 SF-IDCHANGE-TRIGGER 19

5.2.8.1 Description 19
5.2.8.2 SF-IDCHANGE-TRIGGER.request 19
5.2.8.3 SF-IDCHANGE-TRIGGER.confirm 19
5.2.9 SF-ID-LOCK 20
5.2.9.1 Description 20
5.2.9.2 SF-ID-LOCK.request 20
5.2.9.3 SF-ID-LOCK.confirm 20
5.2.10 SF-ID-UNLOCK 20
5.2.10.1 Description 20
5.2.10.2 SF-ID-UNLOCK.request 20
5.2.10.3 SF-ID-UNLOCK.confirm 20
5.2.11 SF-LOG-SECURITY-EVENT 21
5.2.11.1 Description 21
5.2.11.2 SF-LOG-SECURITY-EVENT.request message example 21
5.2.11.3 SF-LOG-SECURITY-EVENT.confirm message example 23
5.2.12 SF-ENCAP 23
5.2.12.1 Description 23
5.2.12.2 SF-ENCAP.request 23
5.2.12.3 SF-ENCAP.confirm 24
5.2.13 SF-DECAP 24
5.2.13.1 Description 24
5.2.13.2 SF-DECAP.request 24
5.2.13.3 SF-DECAP.confirm 25

6 SF-SAP procedures 25
6.1 Outbound message handling 25
6.1.1 Using SF-SIGN, SF-ENCRYPT 25
6.1.2 Using SF-ENCAP 26
6.2 Inbound message handling 26
6.2.1 Using SF-VERIFY and SF-DECRYPT 26
6.2.2 Using SF-DECAP 27
6.3 ID Management 27
6.3.1 IDCHANGE Notifications 27
6.3.1.1 Introduction 27
6.3.1.2 Id-change event hook 28
6.3.1.3 Two phase commit process 28
6.3.2 Prevent IDCHANGES 30
6.3.3 Trigger IDCHANGES 31
6.4 Log security event 31

Annex A (informative): SF-Command 32

A.1 Overview 32

A.2 Description 32
A.2.1 SF-IDCHANGE-EVENT service: SF-COMMAND.request (see clause 5.2.6.2) 32
A.2.2 SF-IDCHANGE-EVENT service: SF-COMMAND.confirm (see clause 5.2.6.3) 33

Annex B (informative): SF-Request 34

B.1 Overview 34

B.2 Description 34
B.2.1 SF-ENCRYPT service: SF-REQUEST.request (see clause 5.2.3.2) 34
B.2.2 SF-ENCRYPT service: SF-REQUEST.confirm (see clause 5.2.3.3) 35

Annex C (informative): Example of service primitives description in the framework of ISO 24102-3 36

C.1 Introduction 36

C.2 Class for SF-SAP Command.request service primitive functions 36

C.3 Class for SF-SAP Command.confirm service primitive functions 36

C.4 Class for SF-SAP Request.request service primitive functions 37

C.5 Class for SF-SAP Request.confirm service primitive functions 37

History 38

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables 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.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

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

The present document is part 9 of a multi-part deliverable. Full details of the entire series can be found in part 1 [i.2].

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

The communications architecture standard ETSI EN 302 665 [i.1], clause 4.4 describes the reference architecture of ITS station, which includes the following internal functional blocks:

- ITS-S Access layer;
- ITS-S Networking & Transport layer;
- ITS-S Facilities layer;
- ITS-S Applications;
- ITS-S Management entity;
- ITS-S Security entity;

and the interfaces between these blocks.

The present document specifies interfaces between the security entity and facilities layer of ITS-S from a functional point of view. Access control to the Service Access Point and further definitions of station internal signals are out of scope of the present document.

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

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 Facilities layer including interface services and service primitives which are extensible in order to achieve general applicability. Additionally, it specifies related procedures and common parameters.

The SF-SAP description in the present document is from a functional point of view according to the ISO model modified by ETSI EN 302 665 [i.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 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 EN 302 665: "Intelligent Transport Systems (ITS); Communications Architecture".

[i.2] ETSI TS 102 723-1: "Intelligent Transport Systems (ITS); OSI cross-layer topics; Part 1: Architecture and addressing schemes".

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

[i.4] ETSI TS 103 097: "Intelligent Transport Systems (ITS); Security; Security header and certificate formats".

[i.5] ETSI TS 102 637-1: "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 1: Functional Requirements".

[i.6] ETSI TS 101 539-2: "Intelligent Transport Systems (ITS); V2X Applications; Part 2: Intersection Collision Risk Warning (ICRW) application requirements specification".

[i.7] ETSI TS 101 539-3: "Intelligent Transport Systems (ITS); V2X Applications; Part 3: Longitudinal Collision Risk Warning (LCRW) application requirements specification".

[i.8] PRESERVE Deliverable D1.3: "V2X Security Architecture V2", January 2014.

NOTE: Available at https://www.preserve-project.eu/www.preserve-project.eu/sites/preserve-project.eu/files/PRESERVE-D1.3-V2X_Security_Architecture_V2.pdf.

[i.9] H. Schweppe, B. Weyl, Y. Roudier, M.S. Idrees, T. Gendrullis, M. Wolf: "Securing car2X applications with effective hardware-software co-design for vehicular on-board networks". In 27th Joint VDI/VW Automotive Security Conference, Berlin, Germany, October 2011. VDI Berichte 2131.

NOTE: Available at https://evita-project.org/Publications/SGIR11.pdf.

[i.10] ETSI EN 302 663: "Intelligent Transport Systems (ITS); ITS-G5 Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band".

[i.11] ETSI EN 303 613: "Intelligent Transport Systems (ITS); LTE-V2X Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band".

[i.12] ETSI TS 101 539-1: "Intelligent Transport Systems (ITS); V2X Applications; Part 1: Road Hazard Signalling (RHS) application requirements specification".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in ETSI EN 302 665 [i.1], ETSI TS 102 940 [1] 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 Symbols

Void.

3.3 Abbreviations

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

SAP	Service Access Point
SF-SAP	Security entity - Facilities layer SAP

4 Architecture

4.1 General

4.1.1 Introduction

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

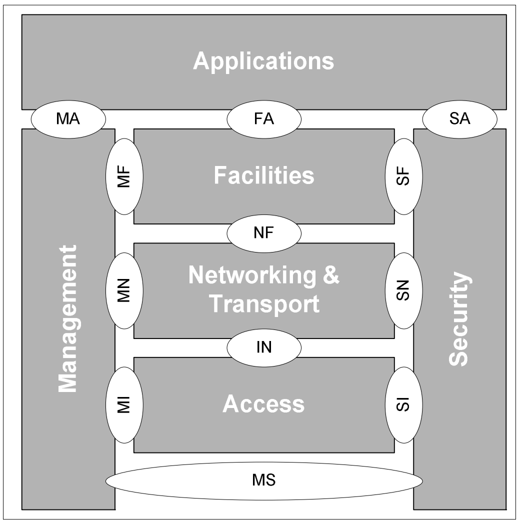

Figure 1: ITS station reference architecture

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.

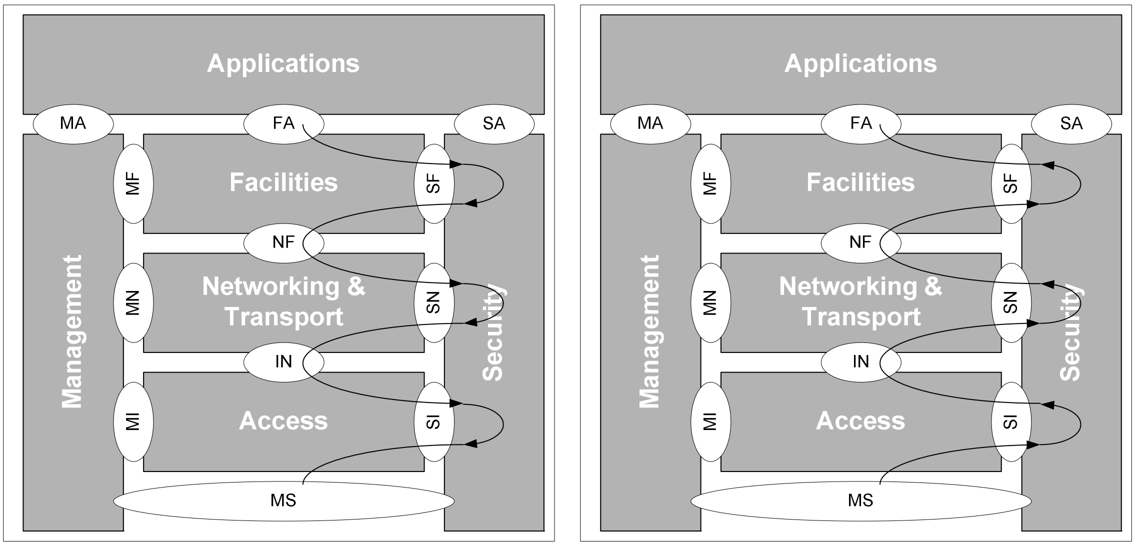

Figure 2: TX (left) and RX (right) information flow through the ITS station

The present document specifies only the SF-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 SF-SAP only. The Security entity acts like a layer inside the Facilities layer, i.e. it is called during the processing of messages traversing the Facilities layer. The security entity will however not act as a layer above or below the Facilities layer. This means that interactions with Applications and Network & Transport layers are achieved via other means, i.e. the FA-SAP is used for the interaction between the Facilities and Applications layers, whereas the NF-SAP is used for the interaction between the Networking & Transport layers and Facilities layers.

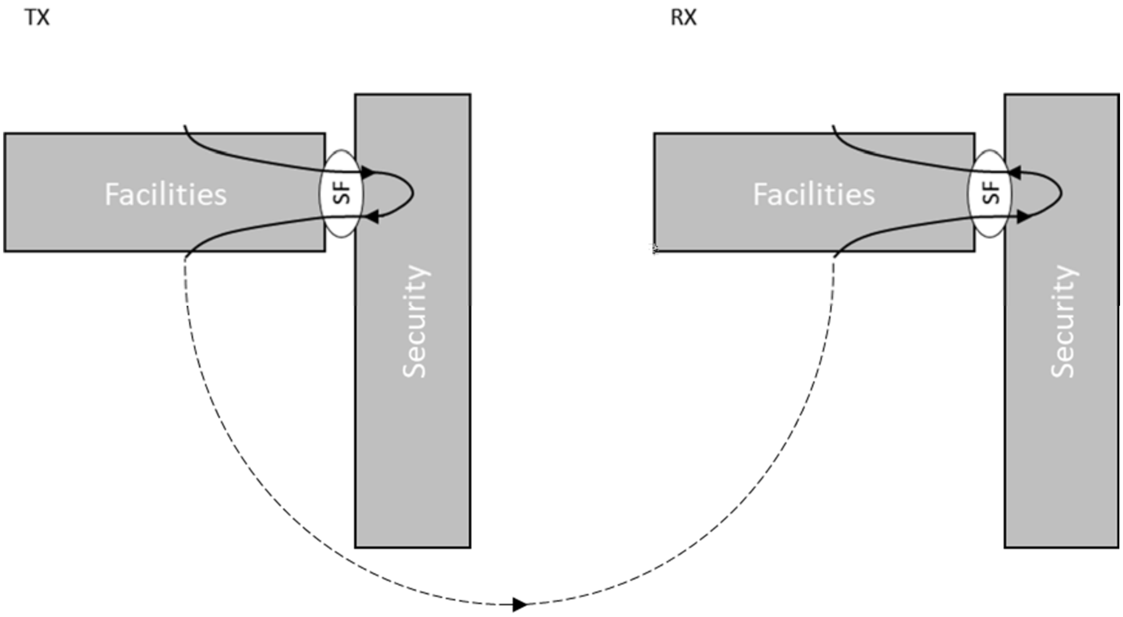

Figure 3: SF-SAP centric Information flow

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, facilities layer 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.

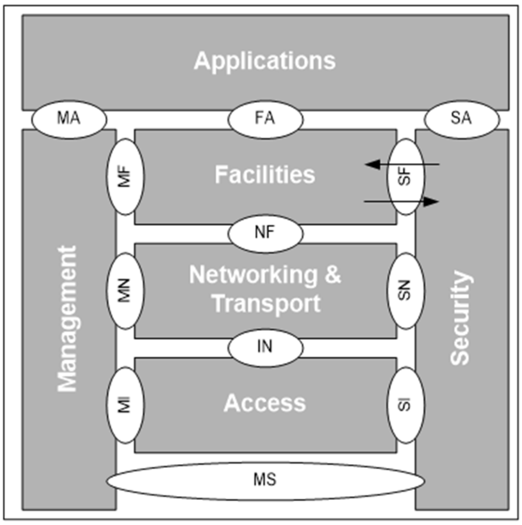

Figure 4: Horizontal Control Communication

4.1.4 Protocol work split

The SF-SAP provides a set of primitive Security functions to the Facilities layer.

Figure 5 shows how a protocol entity within the Facilities layer handles the sending and receiving of information but uses some security extensions to invoke the primitive functions of the Security entity in order to meet the security requirements of this layer. They are supported by the Identity Management Capabilities, specified in ETSI TS 102 940 [1], clause 6, necessary to apply the Atomic Security Capabilities.

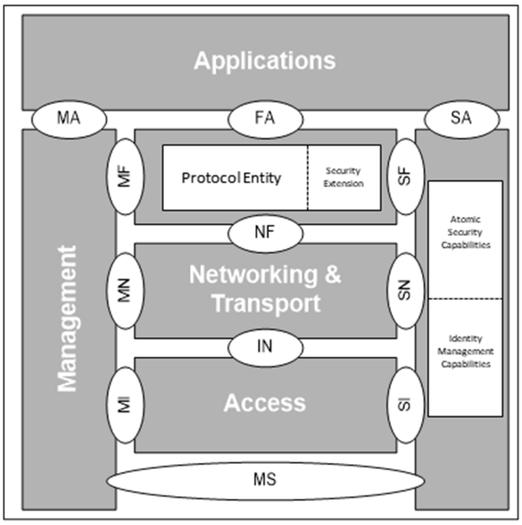

Figure 5: Protocol work split

4.1.5 Multiple instances

The present document does not discuss architecture. However, the SF-SAP can 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 Facilities layer 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. This means that, 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. For instance, a SF-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 [1], clause 5.2. In addition to those, security services used in the research projects PRE-DRIVE C2X and EVITA were adopted and fitted to the existing services. See PRESERVE Deliverable D1.3 [i.8] and [i.9] for documentation on the research project services.

Table 1 summarizes the security services to be specified in the present document, clause 5. These security services shall be invoked directly by applications or other components and layers according to ETSI TS 102 940 [1]. 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	SF-ENCRYPT
	Decrypt Single Message	Request	SF-DECRYPT
Authentication and Integrity	Authorize Single Message	Request	SF-SIGN
	Validate Authorization on Single Message	Request	SF-VERIFY
Identity Management	Lock ID Change	Request	SF-ID-LOCK
	Unlock ID Change	Request	SF-ID-UNLOCK
	Subscribe to ID Change Notification	Request	SF-IDCHANGE-SUBSCRIBE
	Unsubscribe from ID Change Notification	Request	SF-IDCHANGE-UNSUBSCRIBE
	Change ID	Indication send to subscribed entities	SF-IDCHANGE-EVENT
	Trigger ID Change	Request	SF-IDCHANGE-TRIGGER
Extras	Log Security Event	Request	SF-LOG-SECURITY-EVENT
	Extract Permissions	Request	SF-EXTRACT-PERMISSIONS
	Encapsulate Message	Request	SF-ENCAP
	Decapsulate Message	Request	SF-DECAP

5 Interfaces between the Security entity and the Facilities layer

5.1 Interface services

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

- SF-SIGN
  Create authentication information for outgoing ITS messages

- SF-VERIFY
  Validate authentication information from incoming ITS messages

- SF-ENCRYPT
  Encrypt outgoing ITS single messages

- SF-DECRYPT
  Decrypt incoming ITS single messages

- SF-IDCHANGE-SUBSCRIBE
  Subscribe for notifications on SF-IDCHANGE-EVENT, used for concurrent identifiers exchange across the ITS-S

- SF-IDCHANGE-EVENT
  The indication sent to subscribers on IDCHANGE

- SF-IDCHANGE-UNSUBSCRIBE
  Unsubscribe for IDCHANGE notifications, cf. SF-IDCHANGE-EVENT

- SF-IDCHANGE-TRIGGER
  Ask security entity to trigger IDCHANGE procedure

- SF-ID-LOCK
  Ask security entity to avoid IDCHANGEs

- SF-ID-UNLOCK
  Release SF-ID-LOCK

- SF-LOG-SECURITY-EVENT
  Insert external security events
- SF-ENCAP
  Encapsulate outbound messages in a security envelope. This is an alternative way of calling the same functionality that SF-SIGN and/or SF-ENCRYPT offer, where the security parameter selection is done via a security profile parameter or security entity pre-sets
- SF-DECAP
  Decapsulate inbound messages from a security envelope. This is an alternative way of calling the same functionality that SF-VERIFY and/or SF-DECRYPT offer, and should be used together with SF-ENCAP

5.2 Service primitives and parameters

5.2.1 SF-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 SF-SIGN.request

SF-SIGN.request is sent from the Facilities layer to the Security entity for executing the SIGN service. The parameters shall be as described in Table 2.

Table 2: SF-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 Facilities 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.4] 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 SF-SIGN.confirm

SF-SIGN.confirm is sent from the Security entity to the Facilities layer as a corresponding reply to SF-SIGN.request. The parameters shall be as described in Table 3.

Table 3: SF-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 SF-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 SF-VERIFY.request

SF-VERIFY.request is sent from the Facilities layer to the Security entity for executing the VERIFY service. The parameters shall be as described in Table 4.

Table 4: SF-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 SF-VERIFY.confirm

SF-VERIFY.confirm is sent from the Security entity to the Facilities layer as a corresponding reply to SF-VERIFY.request. The parameters shall be as described in Table 5.

Table 5: SF-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	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 Facilities 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.4] security protocol, the permissions contain the SSP associated with ITS-AID	Mandatory

5.2.3 SF-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 SF-ENCRYPT.request

SF-ENCRYPT.request is sent from the Facilities layer to the Security entity for executing the ENCRYPT service. The parameters shall be as described in Table 6.

Table 6: SF-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 SF-ENCRYPT.confirm

SF-ENCRYPT.confirm is sent from the Security entity to the Facilities layer as a corresponding reply to SF-ENCRYPT.request. The parameters shall be as described in Table 7.

Table 7: SF-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 SF-DECRYPT

5.2.4.1 Description

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

5.2.4.2 SF-DECRYPT.request

SF-DECRYPT.request is sent from the Facilities layer to the Security entity for executing the DECRYPT service. The parameters shall be as described in Table 8.

Table 8: SF-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 SF-DECRYPT.confirm

SF-DECRYPT.confirm is sent from the Security entity to the Facilities layer as a corresponding reply to SF-DECRYPT.request. The parameters shall be as described in Table 9.

Table 9: SF-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 SF-IDCHANGE-SUBSCRIBE

5.2.5.1 Description

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

5.2.5.2 SF-IDCHANGE-SUBSCRIBE.request

SF-IDCHANGE-SUBSCRIBE.request is sent from the Facilties layer to the Security entity for executing the IDCHANGE-SUBSCRIBE service. The parameters shall be as described in Table 10.

Table 10: SF-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 SF-IDCHANGE-SUBSCRIBE.confirm

SF-IDCHANGE-SUBSCRIBE.confirm is sent from the Security entity to the Facilities layer as a corresponding reply to SF-IDCHANGE-SUBSCRIBE.request. The parameters shall be as described in Table 11.

Table 11: SF-IDCHANGE-SUBSCRIBE.confirm

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

5.2.6 SF-IDCHANGE-EVENT

5.2.6.1 Description

Indication for notifications on IDCHANGE, see SF-IDCHANGE-SUBSCRIBE specified in clause 5.2.5.

5.2.6.2 SF-IDCHANGE-EVENT.indication

SF-IDCHANGE-EVENT.indication is sent from the Security entity to the Facilities layer for executing the IDCHANGE-EVENT service. The parameters shall be as described in Table 12.

Table 12: SF-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 SF-IDCHANGE-EVENT.response

SF-IDCHANGE-EVENT.response is sent from the Facilities layer to the Security entity as a corresponding reply to SF-IDCHANGE-EVENT.indication. The parameters shall be as described in Table 13.

Table 13: SF-IDCHANGE-EVENT.response

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

5.2.7 SF-IDCHANGE-UNSUBSCRIBE

5.2.7.1 Description

Unsubscription for IDCHANGE notifications, see SF-IDCHANGE-SUBSCRIBE specified in clause 5.2.5.

5.2.7.2 SF-IDCHANGE-UNSUBSCRIBE.request

SF-IDCHANGE-UNSUBSCRIBE.request is sent from the Facilities layer to the Security entity for executing the IDCHANGE-UNSUBSCRIBE service. The parameters shall be as described in Table 14.

Table 14: SF-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 SF-IDCHANGE-UNSUBSCRIBE.confirm

SF-IDCHANGE-UNSUBSCRIBE.confirm is sent from the Security entity to the Facilities layer as a corresponding reply to SF-IDCHANGE-UNSUBSCRIBE.request. The parameters shall be as described in Table 15.

Table 15: SF-IDCHANGE-UNSUBSCRIBE.confirm

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

5.2.8 SF-IDCHANGE-TRIGGER

5.2.8.1 Description

This service is needed in order to ask the Security entity to trigger IDCHANGE procedure.

5.2.8.2 SF-IDCHANGE-TRIGGER.request

SF-IDCHANGE-TRIGGER.request is sent from the Facilities layer to the Security entity for executing the IDCHANGE-TRIGGER service. The parameters shall be as described in Table 16.

Table 16: SF-IDCHANGE-TRIGGER.request

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

5.2.8.3 SF-IDCHANGE-TRIGGER.confirm

SF-IDCHANGE-TRIGGER.confirm is sent from the Security entity to the Facilities layer as a corresponding reply to SF-IDCHANGE-TRIGGER.request. The parameters shall be as described in Table 17.

Table 17: SF-IDCHANGE-TRIGGER.confirm

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

5.2.9 SF-ID-LOCK

5.2.9.1 Description

This service is needed in order to ask the 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 SF-ID-UNLOCK.

5.2.9.2 SF-ID-LOCK.request

SF-ID-LOCK.request is sent from the Facilities layer to the Security entity for executing the ID-LOCK service. The parameters shall be as described in Table 18.

Table 18: SF-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 SF-ID-LOCK.confirm

SF-ID-LOCK.confirm is sent from the Security entity to the Facilities layer as a corresponding reply to SF-ID-LOCK.request. The parameters shall be as described in Table 19.

Table 19: SF-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 SF-ID-UNLOCK

5.2.10.1 Description

This service is used to release SF-ID-LOCK.

5.2.10.2 SF-ID-UNLOCK.request

SF-ID-UNLOCK.request is sent from the Facilities layer to the Security entity for executing the ID-UNLOCK service. The parameters shall be as described in Table 20.

Table 20: SF-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 SF-ID-UNLOCK.confirm

SF-ID-UNLOCK.confirm is sent from the Security entity to the Facilities layer as a corresponding reply to SF-ID-UNLOCK.request. The parameters shall be as described in Table 21.

Table 21: SF-ID-UNLOCK.confirm

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

5.2.11 SF-LOG-SECURITY-EVENT

5.2.11.1 Description

This service is used to insert external security events and is not specified in the present document. An example of a possible set of parameters associated to a SF-LOG-SECURITY-EVENT.request message is shown in clause 5.2.11.2. In this example, a SF-LOG-SECURITY-EVENT.confirm message (clause 5.2.11.3) does not contain any parameters.

5.2.11.2 SF-LOG-SECURITY-EVENT.request message example

SF-LOG-SECURITY-EVENT.request is sent from the Facilities layer to the Security entity for executing the LOG-SECURITY-EVENT service. Table 22 shows a possible set of parameters (and related status) associated to this message based on the example provided in [i.8] but further studies are needed.

Table 22: SF-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 ITS stations, expressed via certificate hash	Mandatory
event_time	INTEGER	0 to 2^{32} - 1 (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 ITS station 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.

- 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.

- 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.

- DISALLOWED_MESSAGE_CONTENT: a message-based plausibility check uses predefined rules and physical boundaries. These checks use a transmitted location data that contains the position of the sender, its current speed and heading at a specific point in time. The values of given location data are compared with the predefined domain of definition. The heading value will have to follow the domain of definition according to related standardization e.g. for CAM and DENM. For example, a heading value larger than 360° should be considered to be not plausible. Furthermore, the velocity values will 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.5].

- REPLAY_DETECTION_TIME: in a time- replay check, the maximum transmission delay will be verified at the receiving station, e.g. according to ETSI TS 101 539-1 [i.12], ETSI TS 101 539-2 [i.6], and ETSI TS 101 539-3 [i.7]. 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 for short-range communication technologies, the distance between a single-hop sender and the own position of the receiver is calculated. For example, if this distance is greater than the maximum transmission range of a radio using the maximum specified transmission power according to ETSI EN 302 663 [i.10] and ETSI EN 303 613 [i.11], 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. For example, 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 for short-range communication technologies 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 will 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 ITS station 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 ITS stations 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 do regularly broadcast 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 ITS stations) or reputation information about their neighbour ITS stations. 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 SF-LOG-SECURITY-EVENT.confirm message example

SF-LOG-SECURITY-EVENT.confirm is sent from the Security entity to the Facilities layer as a corresponding reply to SF-LOG-SECURITY-EVENT.request. In this example, no parameters are associated to this message.

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

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

5.2.12 SF-ENCAP

5.2.12.1 Description

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

5.2.12.2 SF-ENCAP.request

The service primitive SF-ENCAP.request is sent from the Facilities layer to the security entity for executing the ENCAP service. The parameters shall be as described in Table 24.

Table 24: SF-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 Facilities layer 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.4] 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 SF-ENCAP.confirm

The service primitive SF-ENCAP.confirm is sent from the Security entity to the Facilities layer as a corresponding reply to a SF-ENCAP.request. The parameters shall be as described in Table 25.

Table 25: SF-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 SF-DECAP

5.2.13.1 Description

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

5.2.13.2 SF-DECAP.request

The service primitive SF-DECAP.request is sent from the Facilities layer to the Security entity for executing the DECAP service. The parameters shall be as described in Table 26.

Table 26: SF-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 SF-DECAP.confirm

The service primitive SF-DECAP.confirm is sent from the Security entity to the Facilities layer as a corresponding reply to a SF-DECAP.request. The parameters shall be as described in Table 27.

Table 27: SF-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	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 Facilities layer 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 the senders permissions, the DECAP service may report those back to the caller. For example, when using ETSI TS 103 097 [i.4] security protocol, the permissions contain the SSP associated with ITS-AID	Mandatory

6 SF-SAP procedures

6.1 Outbound message handling

6.1.1 Using SF-SIGN, SF-ENCRYPT

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

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

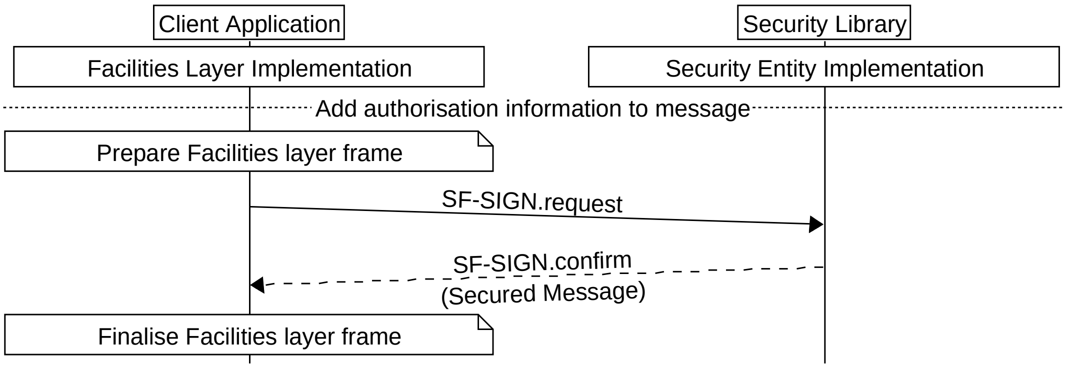
Client Application
Security Library
Facilities Layer Implementation
Security Entity Implementation
Add authorisation information to message
Prepare Facilities layer frame
SF-SIGN.request
SF-SIGN.confirm
(Secured Message)
Finalise Facilities layer frame

Figure 6: Using SF-SIGN

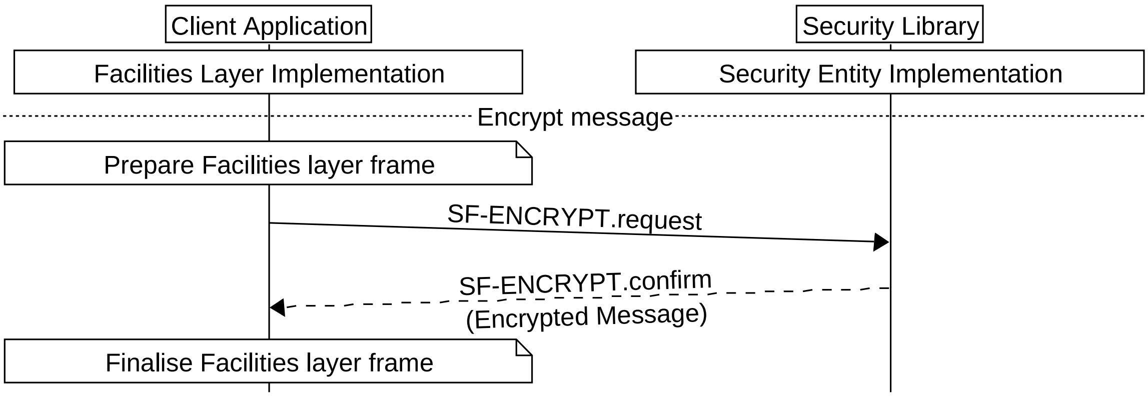
Client Application
Security Library
Facilities Layer Implementation
Security Entity Implementation
Encrypt message
Prepare Facilities layer frame
SF-ENCRYPT.request
SF-ENCRYPT.confirm
(Encrypted Message)
Finalise Facilities layer frame

Figure 7: Using SF-ENCRYPT

6.1.2 Using SF-ENCAP

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

6.2 Inbound message handling

6.2.1 Using SF-VERIFY and SF-DECRYPT

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

When a Facilities layer message is received, the Facilities layer implementation can verify the sender authentication information by using the SF-VERIFY service (see Figure 8) or decrypt encrypted messages using the SF-DECRYPT service (see Figure 9). The distinction can be made e.g. based on the transmission mode, unicast, multicast, or broadcast, since broadcast ITS communications is unencrypted by default.

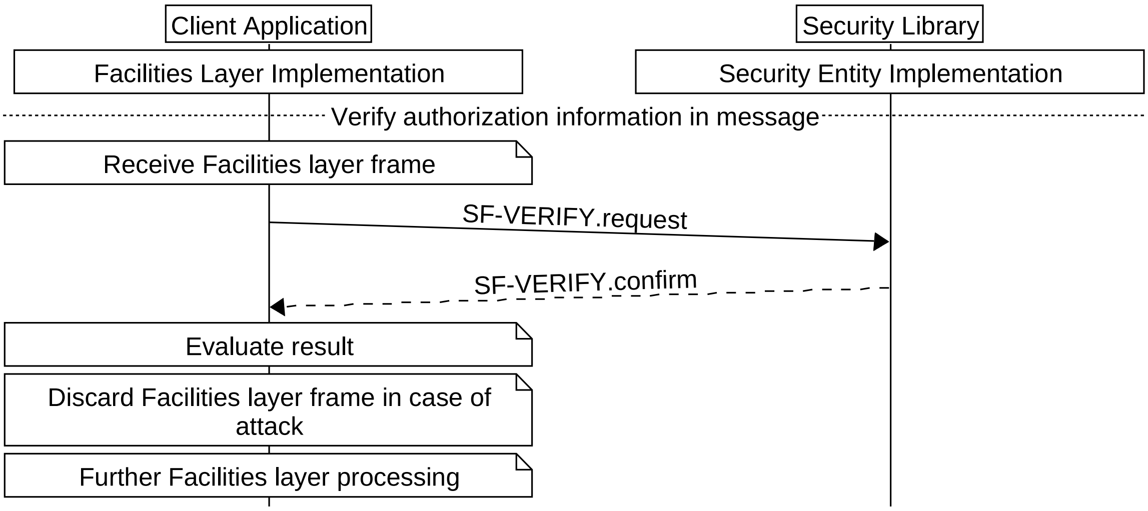
Client Application
Security Library
Facilities Layer Implementation
Security Entity Implementation
Verify authorization information in message
Receive Facilities layer frame
SF-VERIFY.request
SF-VERIFY.confirm
Evaluate result
Discard Facilities layer frame in case of attack
Further Facilities layer processing

Figure 8: Using SF-VERIFY

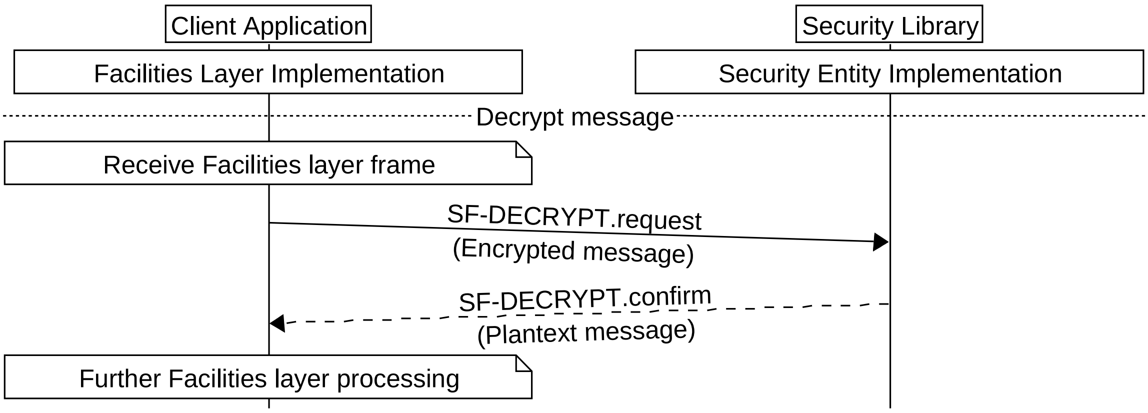
Client Application
Security Library
Facilities Layer Implementation
Security Entity Implementation
Decrypt message
Receive Facilities layer frame
SF-DECRYPT.request
(Encrypted message)
SF-DECRYPT.confirm
(Plantext message)
Further Facilities layer processing

Figure 9: Using SF-DECRYPT

6.2.2 Using SF-DECAP

The SF-DECAP service corresponds to the SF-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 an ITS station 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 SF-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 the Security entity

6.3.1.3 Two phase commit process

1) Subscription
For the Facilities layer subscription see Figure 10.

2) Two phase notification
This is 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
This is outlined in Figure 14.
If a component wants to unsubscribe, it may do so by using the SF-IDCHANGE-UNSUBSCRIBE service.

4) Deregistration
This is outlined in Figure 15.
Deregistration may also be invoked by the security entity.

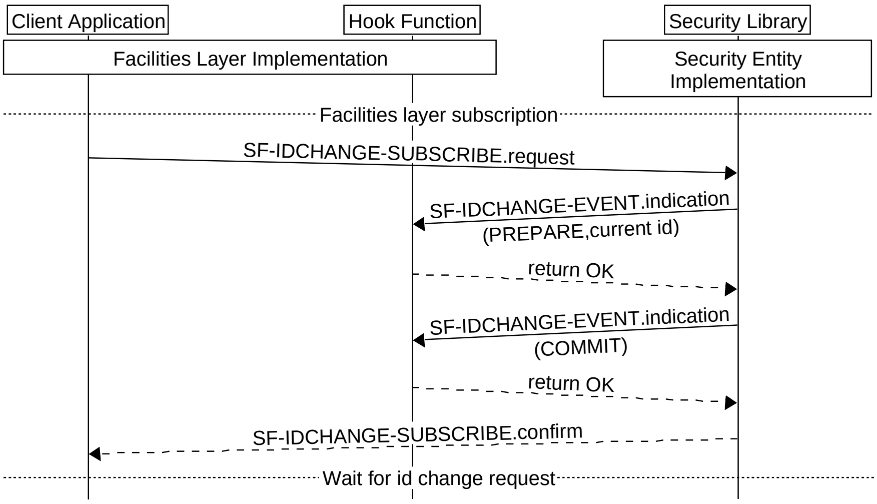
Client Application
Hook Function
Security Library
Facilities Layer Implementation
Security Entity Implementation
Facilities layer subscription
SF-IDCHANGE-SUBSCRIBE.request
SF-IDCHANGE-EVENT.indication
(PREPARE,current id)
return OK
SF-IDCHANGE-EVENT.indication
(COMMIT)
return OK
SF-IDCHANGE-SUBSCRIBE.confirm
Wait for id change request

Figure 10: Subscription

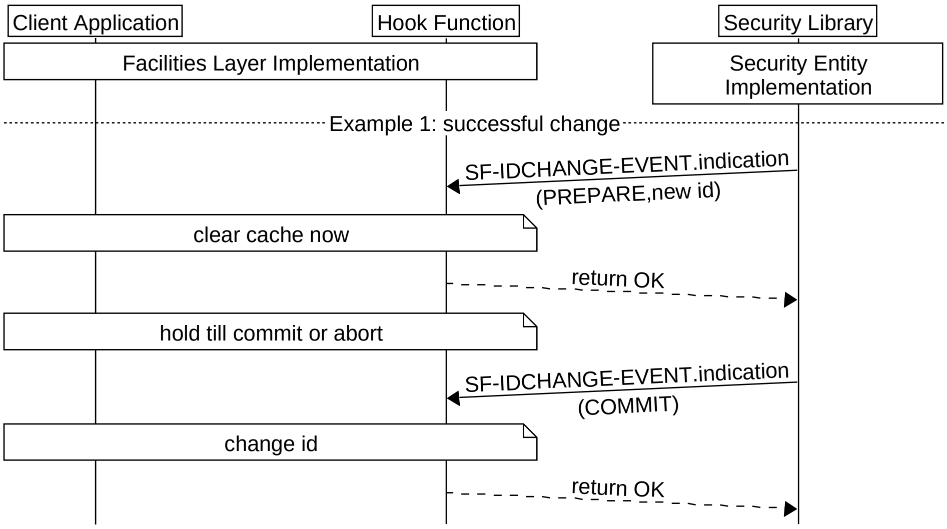
Client Application
Hook Function
Security Library
Facilities Layer Implementation
Security Entity Implementation
Example 1: successful change
SF-IDCHANGE-EVENT.indication (PREPARE,new id)
clear cache now
return OK
hold till commit or abort
SF-IDCHANGE-EVENT.indication (COMMIT)
change id
return OK

Figure 11: Notification and successful change

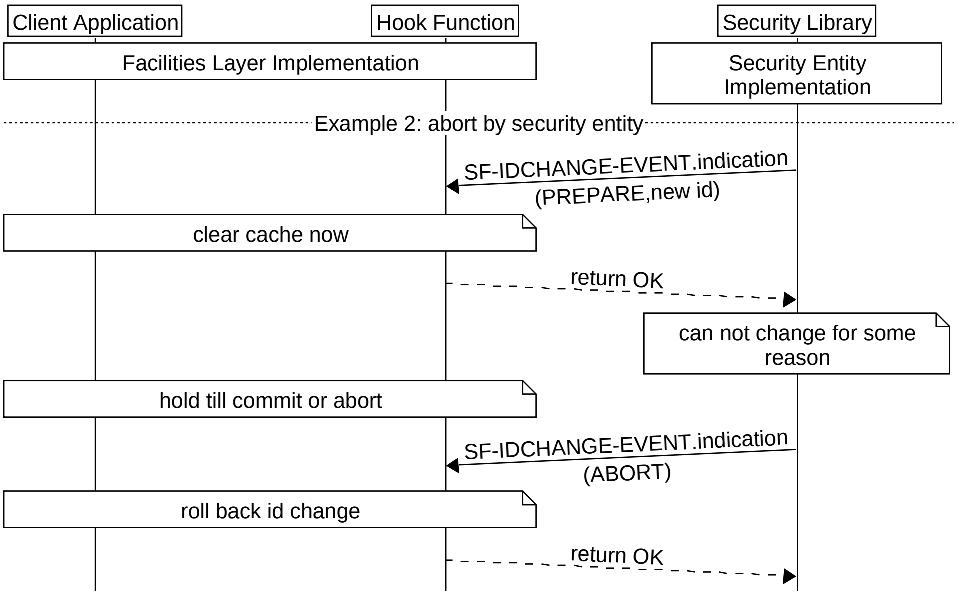
Client Application
Hook Function
Security Library
Facilities Layer Implementation
Security Entity Implementation
Example 2: abort by security entity
SF-IDCHANGE-EVENT.indication (PREPARE,new id)
clear cache now
return OK
can not change for some reason
hold till commit or abort
SF-IDCHANGE-EVENT.indication (ABORT)
roll back id change
return OK

Figure 12: Notification and abort by security entity

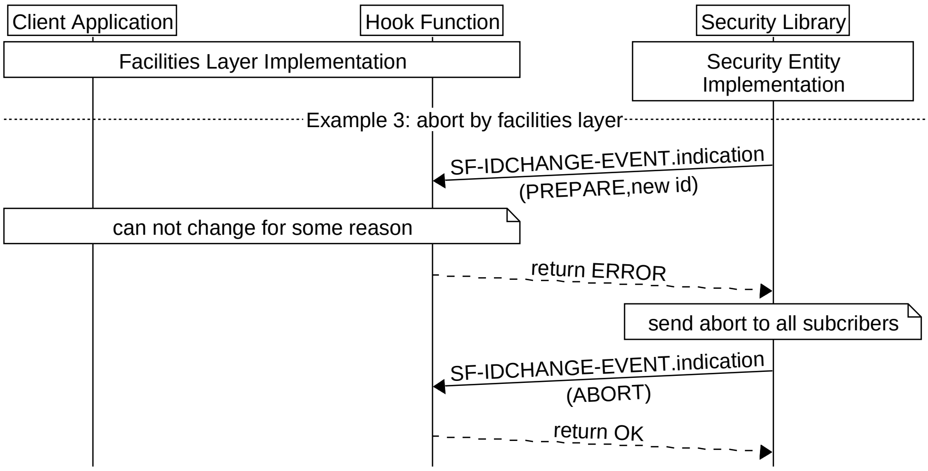
Client Application
Hook Function
Security Library
Facilities Layer Implementation
Security Entity Implementation
Example 3: abort by facilities layer
SF-IDCHANGE-EVENT.indication (PREPARE,new id)
can not change for some reason
return ERROR
send abort to all subcribers
SF-IDCHANGE-EVENT.indication (ABORT)
return OK

Figure 13: Notification and abort by Facilities layer

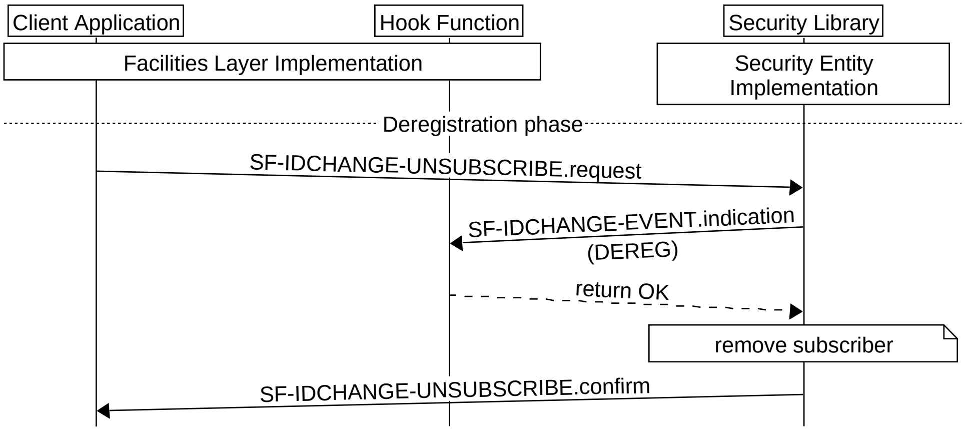
Client Application
Hook Function
Security Library
Facilities Layer Implementation
Security Entity Implementation
Deregistration phase
SF-IDCHANGE-UNSUBSCRIBE.request
SF-IDCHANGE-EVENT.indication (DEREG)
return OK
remove subscriber
SF-IDCHANGE-UNSUBSCRIBE.confirm

Figure 14: Unsubscribe by Facilities layer

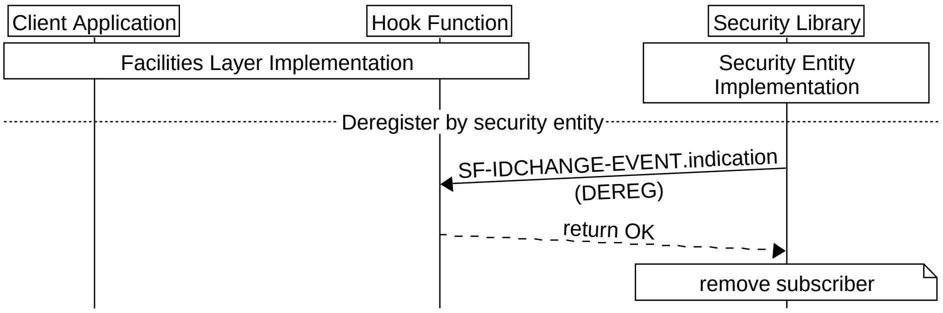
Client Application
Hook Function
Security Library
Facilities Layer Implementation
Security Entity Implementation
Deregister by security entity
SF-IDCHANGE-EVENT.indication (DEREG)
return OK
remove subscriber

Figure 15: Deregistration by Security entity

6.3.2 Prevent IDCHANGES

Call ID-LOCK is used to prevent the Security entity from invoking IDCHANGES. Call ID-UNLOCK is used to unlock. The message 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 ITS station ID.

NOTE 2: Safety applications such as collision avoidance applications ICRW and LCRW (ETSI TS 101 539-2 [i.6], ETSI TS 101 539-3 [i.7]) 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.

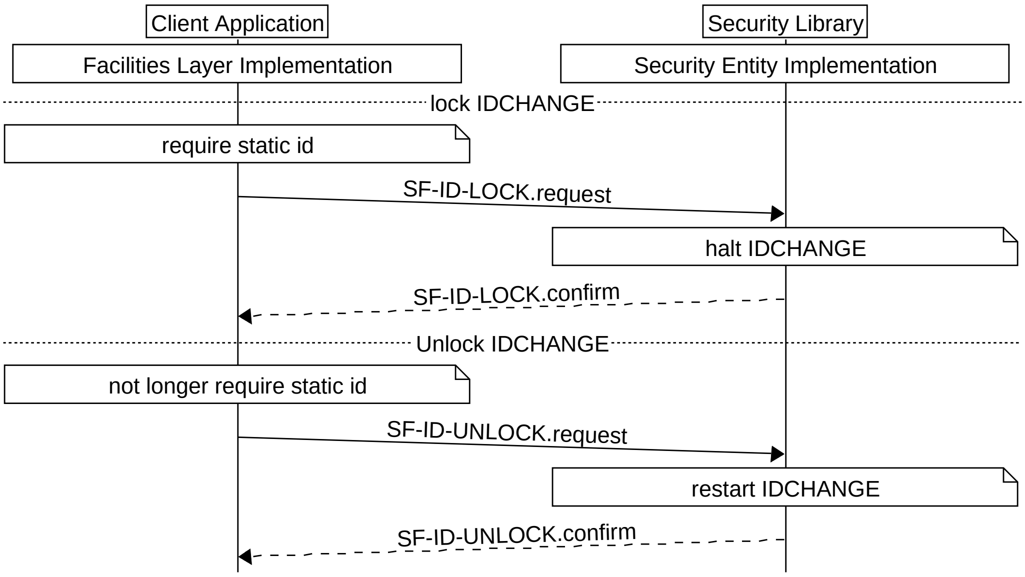

Figure 16: Using ID-LOCK and ID-UNLOCK

6.3.3 Trigger IDCHANGES

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

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

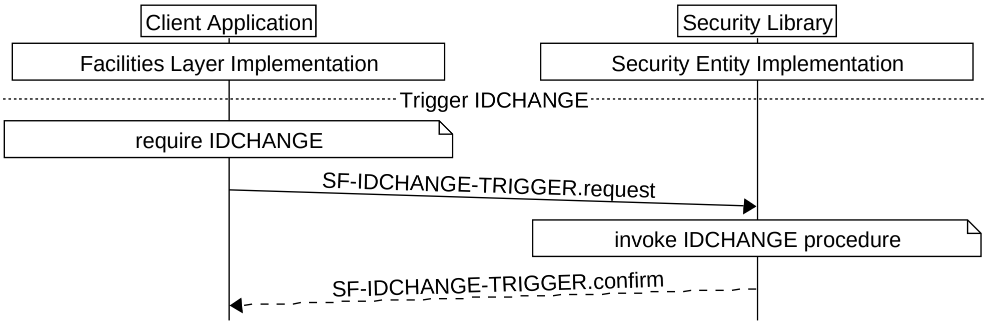
Client Application
Security Library
Facilities Layer Implementation
Security Entity Implementation
Trigger IDCHANGE
require IDCHANGE
SF-IDCHANGE-TRIGGER.request
invoke IDCHANGE procedure
SF-IDCHANGE-TRIGGER.confirm

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 inconsistencies in received messages by the facilities layers.

EXAMPLE 2: 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): SF-Command

A.1 Overview

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

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

Table A.1: SF-Command.No

SF-Command.No	SF-Command Name	Description
0	SF-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 SF-IDCHANGE-EVENT service: SF-COMMAND.request (see clause 5.2.6.2)

Table A.2: SF-COMMAND.request

Name	Type	Valid range	Description	Status
CommandRef	INTEGER	0 to 2^{16} - 1	The cyclic number used as identifier of the SF-REQUEST.request, also used in the corresponding SF-REQUEST.confirm	Mandatory
SF-Command.No	INTEGER	0 to 255	Reference number of the security service SF-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 SF-COMMAND.request identified by the registered value of SF-Command.No of this service.

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

Table A.3: SF-COMMAND.confirm

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

Annex B (informative): SF-Request

B.1 Overview

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

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

Table B.1: SF-Request.No

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

B.2 Description

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

Table B.2: SF-REQUEST.request

Name	Type	Valid range	Description	Status
CommandRef	INTEGER	0 to 2^{16} - 1	The cyclic number used as identifier of the SF-REQUEST.request, also used in the corresponding SF-REQUEST.confirm	Mandatory
SF-Request.No	ENUMERATED	To be fixed later in registration table	Reference number of the security service SF-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 SF-ENCRYPT service: SF-REQUEST.confirm (see clause 5.2.3.3)

Table B.3: SF-REQUEST.confirm

Name	Type	Valid range	Description	Status
CommandRef	INTEGER	0 to 2^{16} - 1	The cyclic number used as identifier of the SF-REQUEST.request, also used in the corresponding SF-REQUEST.confirm	Mandatory
SF-Request.No	ENUMERATED	To be fixed later in registration table	Reference number of the security service SF-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 Introduction

An example implementation of the above service primitive functions in the framework of ISO 24102-3 [i.3] is given below.

C.2 Class for SF-SAP Command.request service primitive functions

```
-- Class for SF-SAP Command.request service primitive functions
SFSAP-CR::=CLASS {
      &mxref RefSFSAP-C UNIQUE,
      &MXParam
      }
-- Named INTEGER constants identify uniquely the functions of the COMMAND service
RefSFSAP-C::=INTEGER {
    c-SF-C-IDCHANGE-EVENT   (0)
    } (0..255)
-- The generic COMMAND.request service primitive
SF-Command-request::=SEQUENCE{
    commandRef  CommandRef, -- see ISO 24102-3 (not related to a specific function)
    ref     SFSAP-CR.&mxref({SF-Command}),
    command-param   SFSAP-CR.&MXParam({SF-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.
SF-Command  SFSAP-CR::={sf-IDCHANGE-EVENT-req, ...}
sf-IDCHANGE-EVENT-req SFSAP-CR::={&mxref c-SF-C-IDCHANGE-EVENT, &MXParam SF-idchange-event-req}
-- Further functions can be added here

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

C.3 Class for SF-SAP Command.confirm service primitive functions

```
-- Class for SF-SAP Command.confirm service primitive functions
-- SFSAP-CC::=CLASS {
      &mxref RefSFSAP-C UNIQUE, -- using the same named INTEGER constants as reference
      &MXParam
      }
-- The generic confirm service primitive
SF-Command-confirm::=SEQUENCE{
    commandRef      CommandRef, -- see ISO 24102-3  (not related to a specific function)
    ref         SFSAP-CC.&mxref({SF-CmdConfirm}),
    cmdConfirm-param    SFSAP-CC.&MXParam({SF-CmdConfirm}{@ref}),
    errStatus       ErrStatus -- see ISO 24102-3  (not related to a specific function)
    }
-- Extendible list of available functions
-- SF-CmdConfirm SFSAP-CC::={sf-IDCHANGE-EVENT-cnf, ...}
sf-IDCHANGE-EVENT-cnf SFSAP-CC::={&mxref c-SF-C-IDCHANGE-EVENT, &MXParam SF-idchange-event-cnf}
-- Further functions can be added here
SF-idchange-event-cnf::=SEQUENCE{

    }
```

C.4 Class for SF-SAP Request.request service primitive functions

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

SF-Request-request::=SEQUENCE{
    commandRef  CommandRef,
    ref     SFSAP-RR.&mxref({SF-Request}),
    request-param   SFSAP-RR.&MXParam({SF-Request}{@ref})
    }
SF-Request SFSAP-RR::={sf-ENCRYPT-req, ...}
sf-ENCRYPT-req SFSAP-RR::={&mxref c-SF-R-ENCRYPT, &MXParam SF-encrypt-req}
SF-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.5 Class for SF-SAP Request.confirm service primitive functions

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

History

Document history		
V1.1.1	March 2021	Publication