UNISIG

| ERTMS/ETCS |
|---|
| **EuroRadio FIS**<br>**Safety Layer** |
| REF : SUBSET-037-2<br>ISSUE : 3.2.8<br>DATE: 13 May 2022 |

| Company | Technical Approval | Management approval |
|---|---|---|
| ALSTOM | | |
| AZD | | |
| CAF | | |
| HITACHI RAIL | | |
| MERMEC | | |
| SIEMENS | | |
| THALES | | |

*© This document has been developed and released by UNISIG*

# 1 MODIFICATION HISTORY

| Issue Number<br>Date | Section Number | Modification / Description | Author |
|---|---|---|---|
| 2.0.0<br>30-March- 2000 | | Final issue to ECSAG | U.D. (ed) |
| 2.1.0<br>23-November-2001 | All | Revision | LK |
| 2.1.7 | | Version with revision marks | LK |
| 2.2.0 | | Final issue after revision | LK |
| 2.2.1 | 3.4, 5.2, 5.3, 7.1.2, 7.2.2, 7.2.4, 7.2.5, 7.3.2, 8.2.3, 8.2.4, 8.3.1, Annex A, B.1, C.1 | Review comments of Unisig super group inserted | LK |
| 2.2.1+ | 7.2.5.3.6, 7.2.5.3.7 | State table updated<br>(state DATA, event DT SaPDU -> splitting in to Conditions Pre 5 and Pre 6; state AR SaPDU, event AR SaPDU -> DI SaPDU added) | TS |
| 2.2.1.++ | 7.3.2.2.1 | Table 23 Bit numbering changed | TS |
| 2.2.2r | 3.4.1.1, 7.2.5.3.6 | Editorial changes | LK |
| 2.2.2 | - | Clean version | LK |
| 2.2.3 | 3., 3.1.1.5, 3.1.16, 3.3, 3.4.1, 7.2.2.2.1.4, 7.2.5.3.7, 8.2.3.1.2, 8.3.1.14 | Review comments of GSM-R users group inserted, Clarifications, references updated | LK |
| 2.2.4 | - | Clean version | LK |
| 2.2.5 | - | Formal release | LK |
| 2.2.5.revA | 3.4, 5.2, 5.7, 7.3.3, 8.2.2, 8.2.4, 8.2.5, 8.3.1 | Proposed changes according to LOP v 020 | LK |
| 2.2.5.revB | 3.4, 5.2, 5.7, 7.3.3, 8.2.2, 8.2.3, 8.2.4, 8.2.5, 8.3.1, B.1, B.5 | Changes of Neu-Ulm meeting | TS+LK |

*© This document has been developed and released by UNISIG*

| | | | |
|---|---|---|---|
| 2.2.5.revC | 5.2.1.7, 7.3.3.5.4, 8.2.2.6, 8.2.2.9, 8.2.3.2.3, 8.3.1.1, 8.3.2.2.1, 8.3.3.1.2, B.1.1.1.9 | Changes of Berlin meeting | LK |
| 2.2.5.revD | 3.4.1, 8.2.5, 8.3.1, 8.3.3, Annex A | Changes of Edinburgh meeting | TS+LK |
| 2.2.5.revE | 3.4, 4.1.1.1, 7.2.2.2.2, 7.2.4.2, 7.3.2, 8.2.2, 8.2.4, Annex D, Annex E | Changes of Stockholm meeting and email discussion | LK |
| 2.2.5.revF | 3.3, 3.4, 7.2.2, 7.2.4, 7.3.2, 8.2.2, 8.2.5, 8.3.1, AnnexD, AnnexE | Changes of Paris meeting and email discussion | WM+LK |
| 2.2.5.revG | 4.1.1.10, 5.71.4, 7.3.3.5.6, 8.2.3.2.5, 8.2.5, 8.2.7, E.2 | Changes of Zürich meeting | PL+LK |
| 2.3.0 | AnnexE.1, Tables 31, 34, 35 | Formal release | JH |
| 2.3.1 | incorporate CR825; insert new Annexes E and F; rename old Annex E to Annex G | | wg |
| 2.3.2 | All; incorporate CR380, CR814, CR970, CR1018; Page setup, layout and references; All § | Changes from meeting July 2011 and review comments | JM/XM |
| 2.3.3-5 | All; editorial | Internal wg reviews | JH |
| 2.3.6 | - | Formal release | ER WG |
| 2.3.7 | | CR1018<br>CR1135 | ER WG<br>XM |
| 2.3.8 | - | Internal WG review | ER WG |
| 2.3.9 | - | Update according SG comments on CR1018<br>CR1137 | XM |
| 3.0.0 | 5.8, 7.2.2, 7.2.3, Table 29, B.7, deleted annex H.<br>Editorial | Baseline 3 release version | ER WG |

*© This document has been developed and released by UNISIG*

| | | | |
|---|---|---|---|
| 3.0.1 | Table 11 | CR1151 | JM |
| 3.0.2 | Front page | Baseline 3 1st maintenance pre-release version | PP |
| 3.1.0 | - | Baseline 3 1st maintenance release version | PP |
| 3.1.1 | | CR741 (Packet data transmission for ETCS) | ER WG |
| 3.1.2 | | Updated with review comments | ER WG |
| 3.1.3 | | Updated with TCP parameters and DNS txt field for comments of EUG | ER WG |
| 3.1.4 | | Updated with comments of the EUG and input from EIRENE.<br>CR1262. | ER WG |
| 3.1.5 | | Updated with comments, multiplexing for mobile interface, KMS | ER WG |
| 3.1.6 | | Updated with comments in Unisig_B3R2_documents-consolidated review sheet | ER WG |
| 3.2.0 | - | Baseline 3 2nd release version | PP |
| 3.2.1 | 8.2.2.7 o) | CR1309 | PL |
| 3.2.2 | 3.1.1.8, Table 29, Table 48, 8.3.2.3, Figure 23, 8.4.2, ANNEX F | CR1146<br>CR1310<br>CR1319 | IH+GR |
| 3.2.3 | 8.4.2.2 Table 44 “k(RX)”<br>8.4.1.8 bullet 8<br>8.4.1.9 4., 5.c.ii) and 7.<br>ANNEX I<br>8.4.1.8 bullet 9<br>8.4.1.9 4., 5.c.iii., 7<br>8.4.2.3.2.2 and 8.4.2.3.3<br>ANNEX I | CR1146<br><br>CR1310<br><br><br><br>CR5049 | SF (on behalf of ER WG) |
| 3.2.4 | 8.4.1.9 7.c.v<br>ANNEX I | CR5049 | ER WG |

© This document has been developed and released by UNISIG

| | | | |
|---|---|---|---|
| 3.2.5 | | Split SS037 to SS037-1 dealing with CFM and SS037-2 dealing with SFM | JS+IH+GR |
| 3.2.6 | | Updated after EECT review | GR |
| 3.2.7 | | Updated after EECT review | GR+FK |
| 3.2.8 | 3.2 3.4 4 5 6.1 6.2 Annex A | Updated after EECT review CR1423 | ER WG |

© This document has been developed and released by UNISIG

# 2 TABLE OF CONTENTS


1 MODIFICATION HISTORY ..... 2
2 TABLE OF CONTENTS ..... 6
3 GENERAL ASPECTS ..... 8
3.1 Scope ..... 8
3.2 Acronyms and abbreviations ..... 9
3.3 Definitions ..... 10
3.4 References ..... 12
4 REFERENCE ARCHITECTURE ..... 13
5 INTERFACE TO SAFE SERVICES ..... 15
5.1 General ..... 15
5.2 Service primitives for safe connection set-up ..... 15
5.3 Service primitives for safe data transfer ..... 18
5.4 Service primitives for connection release ..... 18
5.5 Service primitives for error reporting ..... 20
5.6 Service primitives for mobile network registration ..... 20
5.7 Service primitives for Permitted Mobile Networks ..... 20
6 SAFE FUNCTIONAL MODULE ..... 22
6.1 Service definition ..... 22
6.1.2 Model of the safe services ..... 22
6.1.3 Safe connection set-up ..... 23
6.1.4 Safe data transfer ..... 23
6.1.5 Release of safe connection ..... 23
6.1.6 Error reporting ..... 24
6.2 Safety protocol ..... 25
6.2.1 Introduction ..... 25
6.2.2 Generic MAC-Calculation ..... 25
6.2.3 Functions of the safety layer ..... 25
6.2.4 Time sequences ..... 32
6.2.5 Structure and encoding of safety PDUs ..... 36
6.2.6 State table ..... 39
6.3 Safety Protocol Management ..... 45
6.3.1 Functions of the Safety Protocol Management ..... 45
6.3.2 Configuration Management ..... 45
6.3.3 Supervision and Diagnostics ..... 46
ANNEX A. (NORMATIVE) ASSUMPTIONS PLACED ON THE ATP APPLICATION ..... 51
ANNEX B. (INFORMATIVE) CBC-MAC CALCULATION ..... 52


*© This document has been developed and released by UNISIG*

© This document has been developed and released by UNISIG

# 3 GENERAL ASPECTS

## 3.1 Scope

3.1.1.1 This FIS is applicable to radio communication systems providing communication services for safety-related application processes using open networks. It specifies for ERTMS/ETCS the Radio System Interoperability for message exchange between on-board and trackside equipment in respect to safety-related application processes, like Automatic Train Protection of ETCS level 2/3. Additionally, it specifies for ETCS level 1 the optional message exchange between on-board equipment and radio in-fill unit.

3.1.1.2 In particular this FIS does not define:

- The application functionality and application information flow.
- The open networks used.
- The physical architecture of the radio communication subsystem.

3.1.1.3 Currently, the version handling fixed for ERTMS/ETCS is as follows:

- There is one version of SFM only.

3.1.1.4 Version upgrade for enhanced EuroRadio SFM, if any, will follow the principle as defined in [Subset-026]:

- The on-board SFM may operate with several of its versions.
- The on-board SFM will decide whether it can use the protocol data units (PDUs) received from trackside.
- This version check does not restrict negotiation of connection features by means of safety feature (SFM).

*© This document has been developed and released by UNISIG*

## 3.2 Acronyms and abbreviations

3.2.1.1 For general ERTMS/ETCS terms, definitions and abbreviations refer to [Subset-023]. New terms and abbreviations relevant and used in this FIS are specified here.

| | |
|---|---|
| AR | Authentication Response |
| AU1 | First Authentication message |
| AU2 | Second Authentication message |
| AU3 | Third Authentication message |
| CEPID | Connection EndPoint IDentifier |
| CFM | Communication Functional Module |
| CS | Circuit Switched |
| DA | Destination Address |
| DES | Data Encryption Standard |
| DF | Direction Flag |
| DI | Disconnect |
| DT | Data |
| ETY | ETCS ID type field in a SaPDU |
| ID | Identity |
| IEC | International Electrotechnical Commission |
| ITU | International Telecommunication Union |
| $K_{AB}$ | Authentication Key (same as KMAC) |
| $K_S$ | Session Key (same as KSMAC) |
| KSMAC | Session Key |
| m | message |
| MA | Management |
| MT | Mobile Termination |
| MTI | Message Type Identifier |
| O&M | Operation and Maintenance |
| OSI | Open System Interconnection |
| PDU | Protocol Data Unit |
| PS | Packet Switched |
| QoS | Quality of Service |
| SA | Source Address |
| SaCEPID | Safe Connection EndPoint IDentifier |
| SaF | Safety Features |
| SAP | Service Access Point |

*© This document has been developed and released by UNISIG*

| | |
|---|---|
| SaPDU | Safety Protocol Data Unit |
| SaS | Safety Service |
| SaSAP | Safety Service Access Point |
| SaSDU | Safety Service Data Unit |
| SaUD | Safety User Data |
| SFM | Safe Functional Module |
| TC | Transport Connection |
| TCEPID | Transport Connection EndPoint IDentifier |
| TPDU | Transport Protocol Data Unit |
| TS | Transport Service |
| TSAP | Transport Service Access Point |
| TSDU | Transport Service Data Unit |
| X | Mandatory parameter |
| X(U) | Use of this parameter is a user option |
| X(D) | Use of this parameter is a user option. If not provided, a default value will be used. |

## 3.3 Definitions

3.3.1.1 For general ERTMS/ETCS terms, definitions and abbreviations refer to [Subset-023]. New definitions relevant and used in this FIS are specified here.

**Mandatory feature:** The feature has to be provided by on-board and/or trackside equipment where interoperability is required.

**Optional feature/Option:** The feature might be provided or not. If provided, it has to be provided as specified. Optional features are not required. Interoperability between EuroRadio peers providing and not providing the optional feature has to be guaranteed. Otherwise, the option has to be deactivated.

**National Add-on:**
The feature is a matter of national railway specification. Interoperability must not be influenced.

**CS MODE**

Circuit switched transmission mode uses a dedicated end-to-end transmission resource for each logical connection.

**DATA ENCRYPTION STANDARD (DES)**
A block cipher published in 1977 by the NBS as a US government norm. DES has been renamed Data Encryption Algorithm (DEA) during its adoption as an ANSI standard ([ANSI X3.92], 1981).

**DES KEY**
A cryptographic key of length 64 bits, where each eighth bit is an odd parity bit, as defined in [ANSI X3.92], 1981. Because of this structure, the effective key length is 56 bits.

*© This document has been developed and released by UNISIG*

**DELETION (of a message)**

An attack in which a message is erased from the stream of messages.

**FORM FIT FUNCTIONAL INTERFACE SPECIFICATION (FFFIS)**

A FFFIS is the complete definition of an interface between functional or physical entities.

The FFFIS includes:

- FIS,
- Electrical characteristics related to data,
- communication protocol[1],
- plug.

The FFFIS guarantees the interoperability but not the exchangeability of physical entities.

**FUNCTIONAL INTERFACES SPECIFICATION (FIS)**

A FIS specifies the link between functional modules or between physical entities by:

- The required external data flow,
- The required data characteristics,
- The data range and resolution requirements.

**FUNCTIONAL MODULE**

Set of functions contributing to realize the same global task.

**INSERTION (of a new message)**

An attack in which a new message is being implanted into the stream of messages.

**MESSAGE AUTHENTICATION CODE (MAC)**

An authenticator which is sent with a message to enable the receiver to detect alterations made to the message since it left the sender and to verify that the source of the message is as claimed. The MAC is a function of the whole message and a secret key.

**MODIFICATION (of a message)**

Any unauthorised change of any part of a message.

**PADDING**

The information used to fill the unused part of a message to fill the block size.

**PS MODE**

Packet switched transmission mode shares radio transmission resources between several logical connections.

**RADIO COMMUNICATION SYSTEM**

A radio transmission system providing data communication services via open networks. It can be completed by an safety related transmission system to ensure safe data transmission.

**REPETITION/REPLAY**

An attack in which a message is stored and re-transmitted later.

---

[1]Note that 'Communication protocol' is used with different meanings in the EuroRadio FIS and FFFIS:
In the FIS a communication protocol is a protocol between peer entities within different End Systems connected by a network.
In the FFFIS a communication protocol is a protocol between functional modules or physical entities located in the same End System.

*© This document has been developed and released by UNISIG*

**TRANSMISSION MODE TABLE**

The Transmission mode table contains the transmission mode for each known ETCS ID (i.e. RBC).

**TRIPLE-KEY**

Term used for three concatenated DES-keys, i.e. a length of 192 bits. In this specification, KMAC and KSMAC are both triple-keys.

## 3.4 References

3.4.1.1 This FIS incorporates by dated or undated references, provisions from other publications. The relevant parts of these normative references are cited at the appropriate place in the text and the publications are listed hereafter. For dated references, subsequent amendments to or revisions of any of these publications apply to this FIS only when incorporated in it by amendment or revision. For undated references the latest edition of the publication referred to applies.

| | | |
|---|---|---|
| ANSI X3.92 | 12.80 | American National Standard Data Encryption Algorithm |
| EN 50159 | 09.10 | Safety-Related Communication in Transmission Systems |
| ISO/IEC 9797-1 | 12.99 | Information technology - Security techniques - Messages Authentication Codes (MACs) - Part 1: Mechanisms using a block cipher |
| ITU-T E.212 | 11.98 | The international identification plan for mobile terminals and mobile users |
| ITU-T X.214 | 11.93 | Information Technology - Open System Interconnection - Transport service definition |
| Subset-023 | | Glossary of Terms and Abbreviations |
| Subset-026 | | System Requirements Specification |
| Subset-037-1 | | EuroRadio FIS – CS/PS Communication Function Module |
| Subset-092-2 | | ERTMS EuroRadio Test cases Safety Layer |
| Subset-093 | | GSM-R Interfaces Bearer Service Requirements’ |
| Subset-137 | | On-line Key Management FFFIS |

© This document has been developed and released by UNISIG

# 4 REFERENCE ARCHITECTURE

4.1.1.1 EN 50159 defines the reference architecture for safety-related systems using open transmission systems. The general structure of a safety-related system such as the European Train Control System (Figure 1) is derived from EN 50159.

4.1.1.2 In addition to safety-related information, application processes in the safety-related equipment can exchange non-safety related information with remote application processes using the services of the radio communication system.

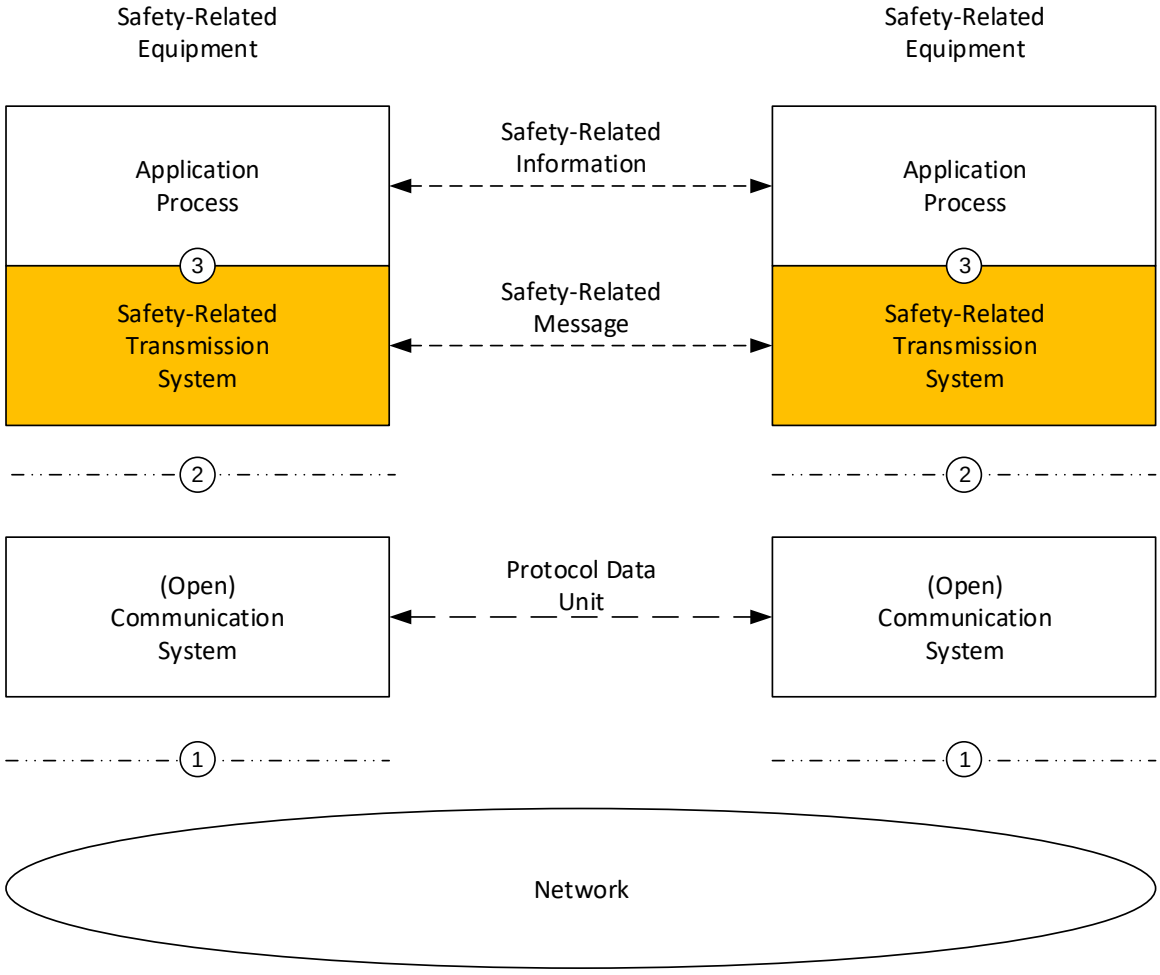



**Figure 1 Structure of the radio communication system**

4.1.1.3 For the purposes of this FIS, the open transmission system of EN 50159 is divided into components: the Communication System and the Open Network. The open (public or railway owned) network is out of scope for this part of the FIS. Only the service features requested at the interface to the network are covered and described in the [Subset-037-1].

4.1.1.4 The Safety Functional Module (SFM) provides the functions of the safety-related transmission system. The Communication Functional Module (CFM) provides the functions of the communication system is covered in the [Subset-037-1]. The service interfaces and the protocol interfaces are defined.

© This document has been developed and released by UNISIG

4.1.1.5 Interface 3 is a service interface between safe applications (e.g. ATP) and the Safe Functional Module (safety layer).

4.1.1.6 Interface 2 is an optional service interface between non-safe applications or support applications and the Communication Functional Module. This option is not required for ETCS level 1 radio in-fill unit.

4.1.1.7 The service interfaces 2 and 3 are not mandatory for interoperability. Only a functional definition is provided.

4.1.1.8 Logical peer entity interface 5 is mandatory for interoperability. The interface is specified in terms of protocol data units and communication relevant aspects of module functionality.

4.1.1.9 The O&M plane covers all operations and management aspects. Interface 4 is a local service interface to the O&M stack, which is not specified.

4.1.1.10 The CFM is out of scope of this FIS.

*© This document has been developed and released by UNISIG*

# 5 INTERFACE TO SAFE SERVICES

## 5.1 General

5.1.1.1 The safe services provided by the SFM are accessed by means of safe service primitives with their corresponding parameters at the SaSAP. The safe service primitives are similar to the service primitives defined in [ITU-T X.214] for connection mode service.

5.1.1.2 The interface is mandatory at functional level only.

5.1.1.3 Note: It is a matter of implementation to adapt this interface to implementation needs and constraints, which do not require any exchange on the air gap and have no impact on the behaviour of the system.

5.1.1.4 Two different types of service primitives are specified:

a) service primitives for safe services

b) service primitives for interworking with the mobile network (out of scope of this FIS)

5.1.1.5 The service primitives for safe services allow the set-up, disconnection of the connection and the safe data transfer.

5.1.1.6 The service primitives for interworking with the mobile network apply to the on-boards only. The service primitives are not safety relevant and have no impact on the safety protocol. They allow the registration to the network and to check the permitted mobile networks for ETCS. These service primitives are applicable if no safe connection exists. These service primitives are described in [Subset-037-1].

## 5.2 Service primitives for safe connection set-up

5.2.1.1 The safe connection set-up service is based on the use of the following primitives:

**Table 1 Service primitives of the safety layer for connection set-up**

| SaS-Primitive<br>Parameter | Sa-CONNECT. request | Sa-CONNECT. indication | Sa-CONNECT. response | Sa-CONNECT. confirm |
|---|---|---|---|---|
| SaCEPID | | X | X(=) | X |
| Called address<br>• Address type<br>• Network address<br>• Mobile Network ID<br>• Called ETCSID type<br>• Called ETCS ID | <br>X<br>X(D)<br>X(U)<br>X<br>X | <br><br><br><br>X<br>X | | |
| Calling address<br>• Calling ETCS ID type<br>• Calling ETCS ID | <br>X(D)<br>X(D) | <br>X(=)<br>X(=) | | |
| Responding address<br>• Responding ETCS ID type<br>• Responding ETCS ID | | | <br>X(D)<br>X(D) | <br>X(=)<br>X(=) |
| Application type | X | X(=) | | |
| Quality of service class | X(D) | | | |

© This document has been developed and released by UNISIG

| |
|---|
| X: mandatory parameter. |
| (=): the value of that parameter is identical to the value of the corresponding parameter of the preceding SaS primitive, if any. |
| X(U) Use of this parameter is an user option |
| X(D) Use of this parameter is an user option. If not provided, a default value will be used. |

5.2.1.2 **SaCEPID:** The local parameter "Safe connection endpoint identifier (SaCEPID)" is provided locally to identify each safe connection at a SaSAP.

5.2.1.3 The **Called address** identifies the called SFM user.

5.2.1.4 The **Address type** qualifies the usage of sub-parameters of called address (refer to [Subset-037-1] Call and ID-Management section for details).

5.2.1.5 The **Network address** contains the network address of the called SaS user. This parameter is composed of sub-fields, e.g. the length of the called number, the type of number, the numbering plan, and the number itself.

5.2.1.6 The **Mobile Network ID** identifies the mobile network. The Mobile Network ID shall consist of the Mobile Country Code and the Mobile Network Code according to [ITU-T E.212].

5.2.1.7 In the case of mobile originated calls, the connection request should contain the sub-parameter Mobile Network ID, to request the appropriate network associated with the called SaS-user.

5.2.1.8 The parameter **ETCS ID type** together with **ETCS ID** is unique within the scope of ETCS and refers to ETCS equipment. The ETCS IDs are used by the safety layer during peer entity authentication. The ETCS-ID type and ETCS ID together with the application type identifies the safety service user.

5.2.1.9 **Called ETCS ID:** The Called ETCS ID parameter conveys the ETCS ID associated with the SaS-user to which the safe connection is to be established.

5.2.1.10 **Calling ETCS ID:** The Calling ETCS ID parameter conveys the ETCS ID of the requesting SaS-user from which the safe connection has been requested.

5.2.1.11 **Responding ETCS ID:** The Responding ETCS ID parameter conveys the ETCS ID of the SaS-user to which the safe connection has been established.

5.2.1.12 **Application type:** The application type is identical at the calling and called side (see section [Subset-037-1] Addressing section).

5.2.1.13 **Quality of Service class:** The QoS parameters give SFM users a method of specifying their needs, and give the CFM a basis for selection of the protocol or for requesting services of lower layers. The QoS class is associated with a set of quality of service parameter values (see [Subset-037-1] QoS parameters section). The service parameter value's applicability can be subject to the transmission mode (CS or PS) for the specific connection. For applicable parameters, the QoS parameters will not be negotiated. The requested and applicable QoS parameter values have to be accepted by the service provider and the peer application, otherwise the connection establishment has to be rejected.

© This document has been developed and released by UNISIG

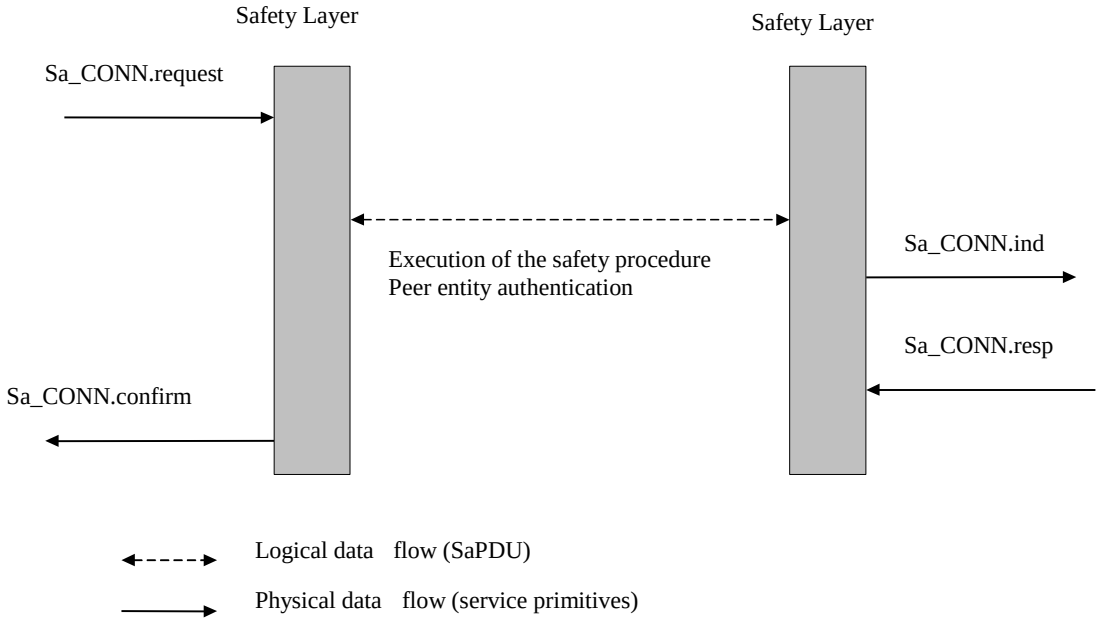



**Figure 2 Sequence of primitives for safe connection set-up**

5.2.1.14 **Sa-CONNECT.request** initiates the establishment of a safe connection. The safety protocol enforces a connection set-up of the underlying transmission system by using T-CONNECT.request.

5.2.1.15 **Sa-CONNECT.indication** is used by the called safety layer entity to inform the called SaS user about the safe connection establishment request.

5.2.1.16 **Sa-CONNECT.response** is used by the responding SaS user to accept the connection to the safety layer entity.

5.2.1.17 **Sa-CONNECT.confirm** is used by the initiating safety layer entity to inform the calling SaS user about the successful establishment of the safe connection after a response of the called SaS user was obtained.

5.2.1.18 Simultaneous requests for safe connection set-up at two SaSAP's are handled independently by the safety layer. These simultaneous requests result in a corresponding number of safe connections. It is the matter of the requesting SaS user to distinguish between confirmations of pending Sa-CONNECT.requests.

*© This document has been developed and released by UNISIG*

## 5.3 Service primitives for safe data transfer

5.3.1.1 For the data transmission two service primitives for the transmission and reception of messages are defined.

**Table 2 Service primitives of the safety layer for data transfer**

| Primitive / Parameter | Sa-DATA.request | Sa-DATA.indication |
|---|---|---|
| SaCEPID | X | X |
| Sa user data | X [1] | X(=) |
| Note1: The length has to be at least 1 octet. | | |

5.3.1.2 Sa-DATA.request on transmission and Sa-DATA.indication on reception perform the safe transfer and the safety procedure ‘message origin authentication’. After the execution of the safety procedure ‘message origin authentication’ the transmitting safety entity forwards the data (user data expanded with a Message Authentication Code) to the transport layer.

5.3.1.3 The user data are transported transparently by the SFM. The recommended size of Sa user data is ≤ 114 octets. The maximum length of SaS user data to be transferred is restricted to 1023 octets.

5.3.1.4 On reception, after successful execution of the procedure ‘message origin authentication’, the user data are delivered to the SaS user using the service primitive Sa-DATA.indication. In the error case, a Sa-REPORT.indication or a Sa-DISCONNECT.indication is delivered.




**Figure 3 Sequence of primitives for safe data transfer**

5.3.1.5 The operation of the safety layer in transferring SaS user data can be modelled as a queue. The ability of a SaS user to issue a Sa-DATA.request depends on the state of the queue. The ability of the safety layer to issue a Sa-DATA.indication depends on the receiving SaS user.

## 5.4 Service primitives for connection release

5.4.1.1 Connection release, i.e. disconnect, is supported by the following two service primitives.

**Table 3 Service primitives of the safety layer for connection release**

| Primitive / Parameter | Sa-DISCONNECT.request | Sa-DISCONNECT.indication |
|---|---|---|

© This document has been developed and released by UNISIG

| | | |
|---|---|---|
| SaCEPID | X | X |
| Disconnect reason | X | X |
| Disconnect sub-reason | X(U) | X |

5.4.1.2 Sa-DISCONNECT.request is used by the SaS user to enforce a release of the safe connection.

5.4.1.3 Sa-DISCONNECT.indication is used to inform the SaS user about a connection release of the safe connection.

5.4.1.4 The reason and sub-reason codes are defined in section 6.3.3.5 ”Error handling”.

5.4.1.5 Normal release requested by a SaS user shall contain the reason code 0; the sub-reason code can be set by the SaS user according to its needs in the range 0...255.

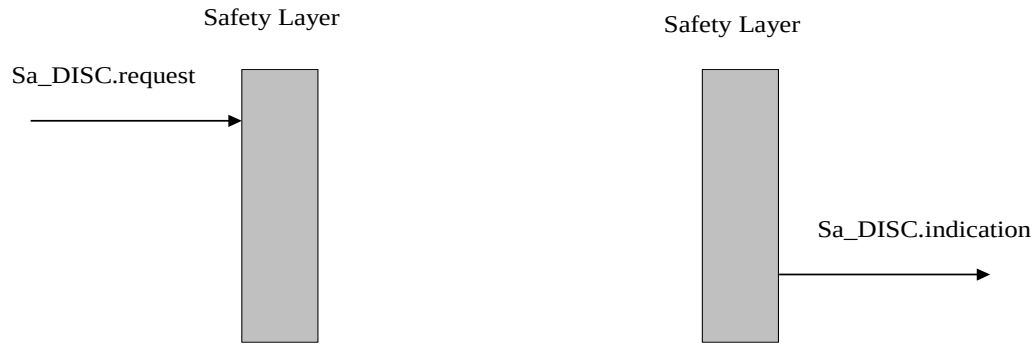



**Figure 4 Sequence of primitives for connection release initiated by a SaS user**

5.4.1.6 The safety layer can issue an unsolicited Sa-DISCONNECT.indication at any time during the connection set-up phase or during the data transfer phase. The release of the connection can be caused by inability of the safety layer to provide a given service.

5.4.1.7 Other sequences of primitives for connection release are possible.

© This document has been developed and released by UNISIG

## 5.5 Service primitives for error reporting

5.5.1.1 Optionally, error reporting is supported by the service primitive Sa-REPORT.indication.

**Table 4 Service primitives for error reporting**

| Primitive<br>Parameter | Sa-REPORT.indication |
|---|---|
| SaCEPID | X |
| Report type | X |
| Number of pairs | X |
| List of pairs | X |

5.5.1.2 The safety layer uses the service primitive Sa-REPORT.indication to inform the SaS user about errors that occur in the safety layer or in the lower layers. The Sa-REPORT.indication is triggered automatically (if the Sa-REPORT.indication is the specified error reaction). The service primitive can be used also for reporting information other than errors (e.g. diagnostics).

5.5.1.3 The parameter **report type** is used to distinguish between the different kinds of information reports. Currently, only report type =1 is defined for error reports.

5.5.1.4 A pair contains two parameters (reason, sub-reason). Refers to section 6.3.3.5 for details about error coding.

## 5.6 Service primitives for mobile network registration

5.6.1.1 Two service primitives are provided for mobile network registration of Mobile Terminations (MT):

- Sa-REGISTRATION.request: to request mobile network registration.
- Sa-REGISTRATION.indication: to indicate mobile network registration status.

5.6.1.2 These service primitives do not provide safe services (i.e. they are not safety relevant and have no impact on the safety protocol).

5.6.1.1 The service primitives are forwarded to/from the Communication Functional Module (CFM) refers to [Subset-037-1] for more details.

5.6.1.2 By means of the service primitive “Sa-REGISTRATION.request” the service user is able to request the registration of one or more Mobile Terminations with one or more mobile networks.

5.6.1.3 The status of registration with mobile networks is indicated by the service primitive “Sa-REGISTRATION.indication” to the service user.

## 5.7 Service primitives for Permitted Mobile Networks

5.7.1.1 It is necessary to indicate a list of 'Permitted' Mobile Networks to the driver. This list comprises mobile networks that are both 'available', i.e. the mobile detects their presence,

*© This document has been developed and released by UNISIG*

and 'Allowed', i.e. a previously-stored list of mobile networks to which the mobile is allowed to register.

5.7.1.2 Two service primitives are provided for indication of allowed mobile networks:

- Sa-PERMISSION.request: to request a list of permitted mobile networks.
- Sa-PERMISSION.indication: to indicate this permitted list.

5.7.1.3 These service primitives do not provide safe services (i.e. they are not safety relevant and have no impact on the safety protocol).

5.7.1.4 The service primitives are command/response between the Communication Functional Module (CFM) and the mobile terminal (MT). Refers to the communication functional module in [Subset-037-1] for more details.

5.7.1.5 By means of the service primitive “Sa-PERMISSION.request” the service user is able to request the indication of permitted mobile networks.

5.7.1.6 The permitted mobile networks are indicated by the service primitive “Sa-PERMISSION.indication” to the service user.

*© This document has been developed and released by UNISIG*

# 6 SAFE FUNCTIONAL MODULE

## 6.1 Service definition

6.1.1.1 The service interface between safety layer user and safety layer is not mandatory for interoperability.

6.1.1.2 This section specifies an interface between the Safe Functional Module (SFM) and the users of the SFM. It gives the data flows to/from the Safe Functional Module, which provides safe services. In the following, the safe service users will be designated by SaS user. The SaS user exchanges data with the SaS provider.

6.1.1.3 The safety services provide safe connection set-up, and safe data transfer during the connection lifetime. The safe data transfer provides data integrity and data authenticity. The SFM reports the errors that occur in the safety layer and transfers error indications from the lower layers.

### 6.1.2 Model of the safe services

6.1.2.1 A safety entity communicates with its users through one or more safe service access points (SaSAP) by means of the safe service primitives. The peer safety entities support safe connection exchanges by means of safety protocol data units (SaPDU). These protocol exchanges use the services of the transport layer via one Transport Connection (TC) through one transport service access point (TSAP), i.e. the safety entity plays the role of a TS user. The exchange of SaPDUs is a logical view only. Service primitives transmit data.

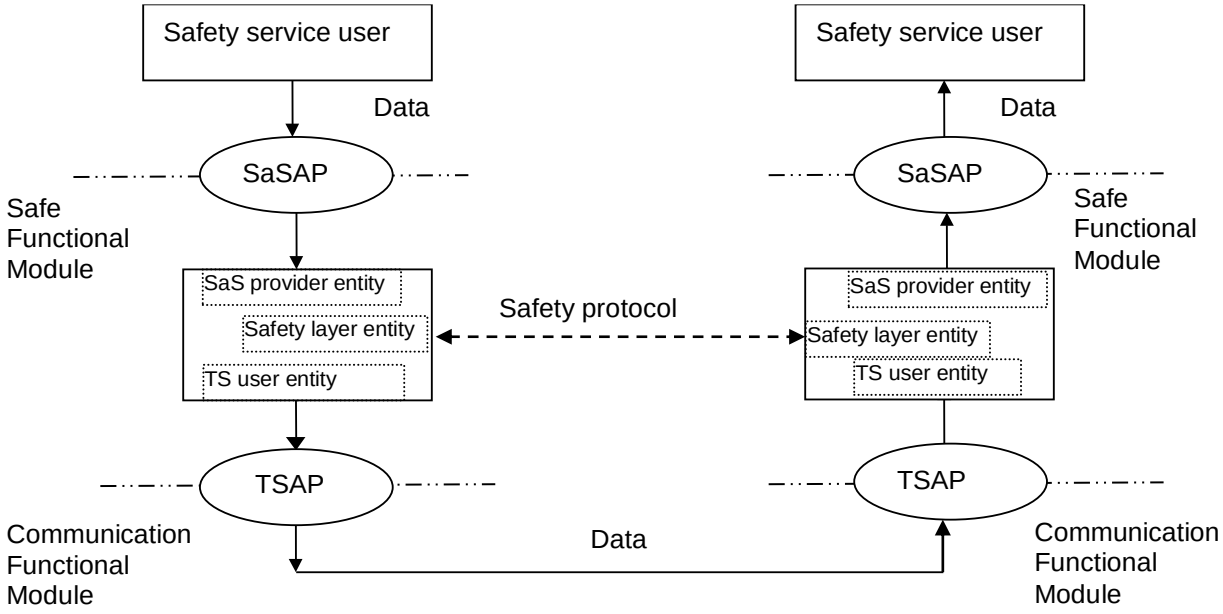



**Figure 5 Model of the safe services**

6.1.2.2 This figure contains a model only. It does not restrict any implementations.

© This document has been developed and released by UNISIG

### 6.1.3 Safe connection set-up

6.1.3.1 Peer entity authentication is provided by the safety protocol between safety layer entities. At connection set-up request, the safety layer will activate the corresponding safety mechanisms to provide entity authentication.

6.1.3.2 The process of establishing a safe connection is initiated at the time when the SaS user requests a connection to the safety layer. The SaS user will send address information and QoS requirements to the safety layer qualifying the request for connection establishment. This QoS value is forwarded to the Communication Functional Module (CFM) and interpreted as a request for a predefined set of quality of service values.

6.1.3.3 The service of providing a safe connection is realised by the execution of the safety procedure ‘peer entity authentication’. The establishment of a transport connection between trackside and trainborne is a precondition for the establishment of the safety connection.

6.1.3.4 Any error in the execution of the safety procedure ‘peer entity authentication’ will result in the rejection of the connection establishment and in the release of the transport connection.

### 6.1.4 Safe data transfer

6.1.4.1 The safety layer provides for an exchange of user data in both directions simultaneously, and preserves the integrity and boundaries of user data.

6.1.4.2 The Safe Functional Module entity guarantees safe data transfer for safety related messages. The safe data transfer service makes use of the safety procedure ‘message origin authentication’.

6.1.4.3 The ‘message origin authentication’ procedure provides a protection against message integrity violation and against insertion of new messages by unauthorised users of the transmission channel. Message integrity violation means any modification of a message from an active attack or due to random transmission channel errors.

6.1.4.4 Each time a SFM entity receives a data message, delivered by the transmission system (the messages coming from SaS users are considered safe), it shall verify that the message was sent by its peer entity, and that the message has not been altered during its transmission. Both operations, i.e. authentication of the sender, and confirmation of message integrity are realised by the execution of the procedure ‘message origin authentication’.

### 6.1.5 Release of safe connection

6.1.5.1 The release of a safe connection is performed by:

a) either or both of the SaS users by releasing an established safe connection;

b) the safety layer by releasing an established safe connection;

c) either or both SaS users by abandoning the safe connection establishment;

d) the safety layer by indicating its inability to establish a requested safe connection.

*© This document has been developed and released by UNISIG*

6.1.5.2 The release of a safe connection is permitted at any time regardless of the current safe connection phase. A request for a release cannot be rejected. The safe service does not guarantee delivery of any Sa user data once the release phase is entered.

6.1.5.3 The request by the SaS user for the release of a safe connection does not need specific safety protection unlike safe connection set-up, because the release of the connection impacts only on availability. In addition, a safe connection is meaningful only if the underlying connections of the lower layers are not released, and a transport or network connection can be released independently from the safety layer.

### 6.1.6 Error reporting

6.1.6.1 The safety layer provides an error reporting function to the SaS user for the established safe connection. Errors occurring are either indicated by the release of the safe connection or optionally by an error report. The inability of the safety layer to provide a service will be reported to the SaS user.

*© This document has been developed and released by UNISIG*

## 6.2 Safety protocol

### 6.2.1 Introduction

6.2.1.1 This section provides a precise specification of the safety protocol taking into account the CENELEC standard EN 50159. The method used in the SFM corresponds to the A1 type in EN 50159: cryptographic safety code using secret key.

### 6.2.2 Generic MAC-Calculation

6.2.2.1 The computation of the MAC in all cases is according to [ISO/IEC 9797-1]. The block cipher used is the single DES with modified MAC algorithm 3, where the last data block in the MAC computation will be computed as encipher with K1, decipher with K2, then encipher with K3 (this is a modification of ISO 9797-1 which uses only two keys, K and K''). ISO 9797-1 Padding Method 1 is used.

6.2.2.2 The CBC-MAC is a value of 64 bits calculated on a message "*m*" using three 64-bit DES keys.

6.2.2.3 To calculate the CBC-MAC on a value X, the length in bits of the value must be a multiple of 64. If necessary, i.e. if the length of a message *m* in bits is not a multiple of 64, padding is performed prior to the computation of the CBC-MAC. As few zero bits as needed (possibly none) are added at the end of the message *m* to obtain a multiple of 64 bits. The padding data *p* is used for CBC-MAC calculation only. It does not become part of the message.

6.2.2.4 The CBC-MAC (K, X) function using a secret triple-key K and the value X = *m* | *p* is defined as follows:

6.2.2.5 Let K = K1 | K2 | K3 be a triple-key and K1, K2, K3 its DES-keys, let X be constituted by the 64-bit blocks $X_1$ | $X_2$ | ... | $X_q$. Let E(Kn,X) be a block cipher function, single DES in CBC mode, enciphering the data string X using the key Kn (n ϵ {1,2,3}). Let $E^{-1}$(Kn,X) be a single DES block decipher function, deciphering the data string X using the key Kn (n ϵ {1,2,3}). Let ⊕ be the XOR-operation. Then, CBC-MAC is derived by the following iteration:

6.2.2.6 The initial value $H_0$ is of length 64 bits, all bits are of value "0". $H_0$ is not enciphered before first usage,

6.2.2.7 $H_i = E(K1,H_{i-1} \oplus X_i)$, i = 1,2,…, q-1, $H_q = E(K3,E^{-1}(K2, E(K1,H_{q-1} \oplus X_q)))$

6.2.2.8 The CBC-MAC calculated on the message *m* is then equal to $H_q$.

6.2.2.9 An informative example is given in ANNEX B.

### 6.2.3 Functions of the safety layer

6.2.3.1 The safety layer provides the safe transfer of user data. This includes the establishment and release of the safety connection.

6.2.3.2 **Safety procedures**

#### 6.2.3.2.1 Message origin authentication / Message integrity

© This document has been developed and released by UNISIG

6.2.3.2.1.1 Message origin authentication/message integrity is a safety procedure ensuring the integrity and authenticity of messages during transmission. It is used to protect the messages against modification and to ensure that no one can masquerade as the originator of the message. In the following, the procedure is simply called message origin authentication because message origin authentication automatically provides message integrity.

**Procedure 1: Message Origin Authentication (MAC) on Transmission (m, $K_S$)**

**Input:** Message m and cryptographic triple key $K_S$, which is shared between the sender (with the source address SA) and the receiver (with the destination address DA); SA and DA are ETCS Identities.

**Procedure:** 1.) Set direction flag of message m (value '0' for initiator, value '1' for responder).

2.) Append the destination address (DA) in front of the message m: "DA | m".

3.) Compute length $\ell$ of string "DA | m" in octets and append length (2 octets [2]) in front of the string for MAC computation, i.e. $\ell$| DA | m

4.) If the length of the message ($\ell$| DA | m) in bits is not a multiple of 64 then perform padding as defined below for $\ell$ | DA | m and append padding data p: ($\ell$| DA | m | p)

5.) Compute MAC for the string " $\ell$| DA | m | p" using the CBC-MAC function and the cryptographic triple key Ks:

MAC(m)=CBC-MAC($K_S$, $\ell$| DA | m | p), where | denotes concatenation

**Output:** If no error occurs MAC(m), which is appended to m. Otherwise, inform the error management.

6.2.3.2.1.2 Message origin authentication is performed as follows:

6.2.3.2.1.3 On transmission of a Data (DT) SaPDU, a Management (MA) SaPDU, the second authentication message (AU2) SaPDU, the third authentication message (AU3) SaPDU, or the Authentication Response (AR) SaPDU, a MAC of length 64 bit is computed using the message m and the cryptographic triple key $K_S$ as input.

6.2.3.2.1.4 For these SaPDUs, the cryptographic triple key $K_s$ used for the computation of the MAC is a session key derived during connection set-up. In addition, in the case of a management SaPDU the triple key $K_s$ is the session key derived during connection set-up. The length of the triple key $K_s = (K_1, K_2, K_3)$ has to be 192 bits including parity bits. In order to get three 64-bit DES-keys for the single DES with modified MAC algorithm 3 from the three 64-bit session key generation outputs, each eighth bit of the 192-bits should be set to an odd-parity value as defined in the standard [ANSI X3.92]. However, setting the parity bits is an implementation matter where the key is internal to an equipment.

[2] The bits in the two octets are numbered from 16 to 1, where bit 1 is the lowest order bit.

© This document has been developed and released by UNISIG

6.2.3.2.1.5 The ETCS Identity of the receiver (DA) is appended before the message "m" for the MAC computation. The Identity is binary coded by 24 bits. If the address is shorter, bits set to zero are added before the address to obtain a receiver identity (DA) of 24 bits.

6.2.3.2.1.6 The length $\ell$ of the string "DA **|** m" is computed and appended before the string "DA **|** m" for the MAC computation. The length $\ell$ is binary coded by 16 bits (without sign) and is not transmitted because the receiver can compute it.

6.2.3.2.1.7 The CBC-MAC ($K_S$, $\ell$ **|** DA **|** m) is then calculated according to the algorithm described in section 6.2.2. If padding is performed prior to MAC calculation, the padding data p is not transmitted because the receiver can compute them, knowing the padding algorithm used.

6.2.3.2.1.8 In the case of a DT SaPDU the message m = '000' **|** MTI **|** DF **|** SaUD consists of the message type identifier (MTI) indicating a DT SaPDU, the direction flag (DF), and the Safety-User Data SaUD.

6.2.3.2.1.9 Concerning the AU2 SaPDU, the message m = ETY **|** MTI **|** DF **|** SA **|** SaF **|** auth2 consists of the ETCS ID type, the message type identifier (MTI) indicating AU2 SaPDU, the direction flag (DF), the source address (SA), the safety features (SaF) and the corresponding authentication message auth2 = "Ra **|** Rb **|** B".

6.2.3.2.1.10 Concerning the AU3 SaPDU, the message m = '000' **|** MTI **|** DF **|** auth3 consists of the message type identifier (MTI) indicating AU3 SaPDU, the direction flag (DF), and the corresponding authentication message auth3 = Rb **|** Ra.

6.2.3.2.1.11 In the case of the AR SaPDU the message m = '000' **|** MTI **|** DF consists of the message type identifier (MTI) indicating the AR SaPDU and the direction flag (DF).

6.2.3.2.1.12 The direction flag is used as a protection against reflection attacks. It is initialised during connection set-up. Its value is zero when the initiator transmits a message and one when the responder of the connection transmits a message.

6.2.3.2.1.13 If an error occurs during the MAC computation the error management is informed and takes over further actions. If no error occurs the output of the MAC computation is the MAC of the message m to be transmitted.

**Procedure 2: Message Origin Authentication (MAC) on Reception (m, $K_S$, MAC'(m'))**

**Input:** Message m including a direction flag, cryptographic triple key $K_S$ which is shared between the sender and receiver (DA is the identity of the receiver), and MAC'(m'), which is the MAC computed for m' by the sender.

**Procedure:** 1.) Append the destination address (DA) in front of the message m: "DA | m".

2.) Compute length $\ell$ of the string (DA | m) in octets and append length (2 octets [3]) in front of the string for MAC computation, e.g. " $\ell$ | DA | m".

[3] The bits in the two octets are numbered from 16 to 1, where bit 1 is the lowest order bit.

© This document has been developed and released by UNISIG

3.) If the length of the message ($\ell$ | DA | m) in bits is not a multiple of 64 then perform padding as defined above for $\ell$ | DA | m and append padding data p;" ($\ell$ | DA | m | p)

4.) Compute MAC for the string ($\ell$ | DA | m | p) using the CBC-MAC function and the cryptographic triple key $K_S$ : CBC-MAC($K_S$, $\ell$ | DA | m | p)

5.) Compare MAC with MAC'.

6.) Verify the value of the direction flag

**Output:** Message m is forwarded to the SaS-user if MAC = MAC' and the value of the direction flag is correct. Otherwise, inform the error management.

6.2.3.2.1.14 On reception of a DT SaPDU, an MA SaPDU, an AU2 SaPDU, an AU3 SaPDU, or an AR SaPDU, a MAC is computed in a similar way to the transmission case. The input parameters are the message m, the cryptographic triple key $K_S$ and the MAC transmitted as part of the received SaPDU. The receiver of the message uses the same parameters, i.e. cryptographic key and algorithms, as the transmitter of the message, derived from the sender and receiver identities and the type of message. The message m consists of the same parts as described above. The receiver adds its ETCS identity (DA) and computes the length $\ell$ of the string "DA **|** m" which has to be added before the message m for the MAC computation and the padding data p, if necessary.

6.2.3.2.1.15 If this MAC for " $\ell$ **|** DA **|** m **|** p" is equal to the MAC transmitted as part of the SaPDU and if the value of the direction flag is correct the user data are forwarded to the SaS-user. If an error occurs, e.g. the value of the direction flag is invalid, the MACs are not equal or there exists no cryptographic key for the underlying connection, the error management is informed and takes over further actions. Normally the evaluation starts with checking the MAC and only if it is correct is the information in the PDU used. The AU2 is an exception to this rule since some of the information inside the PDU is needed to calculate the MAC.

#### 6.2.3.2.2 Peer Entity Authentication

6.2.3.2.2.1 Peer entity authentication is a safety procedure, which is used during connection set-up to compute the session key.

**Procedure 3: Peer Entity Authentication (ETCS ID A, ETCS ID B, $K_{AB}$)**

**Input:** ETCS ID of A and B, authentication triple key ($K_{AB}$) shared between A and B.

**Procedure:** Peer Entity Authentication Protocol as defined in Figure 6

**Output:** In the non error case: successful authentication of A and B against each other, and a session triple key which A and B share

**Error case**: No safety connection between A and B, and the error management is informed

© This document has been developed and released by UNISIG

6.2.3.2.2.2 Peer entity authentication is performed during connection set-up. Its input parameters are the ETCS IDs of A and B which are authenticated against each other and the authentication $_{\text{triple key}}$ $K_{AB}$ shared between A and B. The ETCS IDs of A and B are unique identifiers. The authentication key has been previously established between A and B using a logical or physical key establishment mechanism.

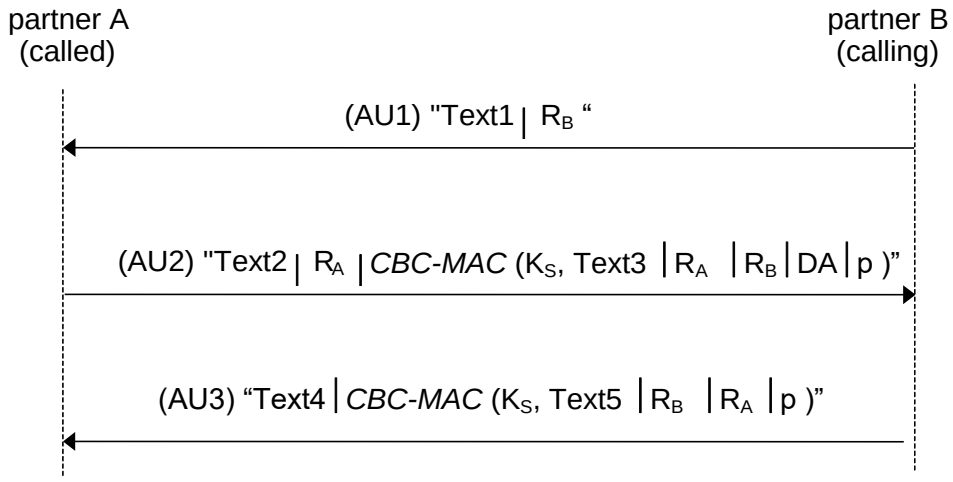



**Figure 6 Sa-Protocol used for peer entity authentication and key generation**

6.2.3.2.2.3 The initiator B of the connection set-up starts the safety association (SA) protocol (see Figure 6) when requesting a transport connection. For the computation of the MAC it makes use of the message origin authentication procedure.

6.2.3.2.2.4 The initiator B transmits a random number $R_B$ of length 64 bits which is generated by B as part of the first authentication message AU1SaPDU to his communication partner A. The random number $R_B$ must be stored (dedicated to the link) before sending AU1SaPDU. After receiving this message, A generates as part of a second authentication message AU2 SaPDU, a random number $R_A$ of length 64 bits, and a MAC computed over the text field text3, the two random numbers $R_A$ and $R_B$, the identity of B (in this context B is the calling ETCS ID) and padding bits. For the computation of the MAC the session key $K_S$ is computed using the session key generation function as described in section 6.2.3.2.3 and the parameters $R_A$, $R_B$ and the authentication key $K_{AB}$. After receiving the message AU2 SaPDU and deriving the key $K_S$, B checks the correctness of the second authentication message received from A. Then, B computes a MAC over the text field text5, and the two random numbers $R_A$ and $R_B$ and transmits it as part of AU3 SaPDU to A. Finally, A checks AU3 SaPDU using the triple key $K_S$.

6.2.3.2.2.5 The fields:

text1 = "ETY | MTI | DF | SA | SaF", where SA = calling ETCS ID,

text2 = "ETY | MTI | DF | SA | SaF", where SA = responding ETCS ID,

text3 = " ℓ | DA | ETY | MTI | DF | SA | SaF",

where DA = calling ETCS ID and SA = responding ETCS ID,

text4 = " '000' | MTI | DF",

© This document has been developed and released by UNISIG

text5 = " ℓ | DA | ‘000’ | MTI | DF", where DA = responding ETCS ID

consist of the ETCS ID type (ETY), the message type identifier (MTI) indicating an authentication SaPDU, the direction flag (DF), the source address (SA) (ETCS Identity on 24 bits), the destination address (DA) (ETCS Identity on 24 bits), and the safety feature SaF.

6.2.3.2.2.6 If no error occurs the output of the peer entity authentication procedure is a successful authentication of A and B against each other and a session key, which is shared between A and B. If an error occurs during the peer entity authentication procedure, then the error management is informed and takes over. No safety connection is established between A and B in this case.

#### 6.2.3.2.3 Cryptographic Keys

6.2.3.2.3.1 Note: key management activities are the matter of other UNISIG Subsets.

6.2.3.2.3.2 The following table describes a three level key hierarchy.

**Table 5 Extended key hierarchy**

| Level | Purpose |
|---|---|
| 3 Transport keys (KTRANS) | Protection of management communication between KMC and RBC or train for establishment or revocation of authentication keys. |
| 2 Authentication keys (KMAC) | Session key derivation in connection establishment. |
| 1 Session keys (KSMAC) | Protection of data transfer between safety entities. |

6.2.3.2.3.3 The level 3 keys (KTRANS) are used by the Key Management Centre to distribute level 2 keys or to change key assignments permanently, including revocation of keys and the introduction of new entities. The Key Management Centre shares a transport key with each entity.

6.2.3.2.3.4 The level 2 keys (KMAC; also referred as $K_{AB}$) are used for session key derivation. Authentication keys (KMAC keys) are level 2 keys, which have been assigned to particular entities. Two entities sharing a common level 2 key can set up a safety association.

6.2.3.2.3.5 The key validity period shall be checked using UTC time and only before establishing a safe connection with a peer entity

6.2.3.2.3.6 Note: management of UTC time (for example derivation and unavailability) is an implementation matter.

6.2.3.2.3.7 Note: if the validity period expires while a safe connection is established, this will not lead to connection release.

6.2.3.2.3.8 The length of a level 2 triple key has to be 192 bits including parity bits, consisting of three 64-bit DES-keys for the single DES with modified MAC algorithm 3.

6.2.3.2.3.9 The level 1 keys (KSMAC; also referred as $K_S$) are derived during peer entity authentication by use of level 2 keys. They are used for the protection during connection

© This document has been developed and released by UNISIG

set-up and data transfer, i.e. MAC computation, in a single session only. They are connection specific and can only be shared by entities that share an authentication key (KMAC key).

6.2.3.2.3.10 Session keys (KSMAC) are DES triple keys, which are used symmetrically, i.e. for both communication directions.

6.2.3.2.3.11 The length of a level 1 triple key is equal to 192 bits consisting of three 64-bit DES-keys.

6.2.3.2.3.12 Session keys are generated using the key derivation function as described in the section below. Both communication partners contribute with their 64-bit (pseudo) random number to the session key.

6.2.3.2.3.13 During the peer entity authentication a session key is derived between two communicating entities using the common authentication triple key KMAC = ($K_1$, $K_2$, $K_3$) of these entities. One 192-bit KSMAC triple key shall be generated by the key derivation procedure. The derivation of the corresponding DES session keys is specified as follows between entities A and B:

6.2.3.2.3.14 The random numbers $R_X$ ($X \in \{A,B\}$) are split into a left ($R_X^L$) and a right ($R_X^R$) 32-bit block:

$$R_A = R_A^L \mid R_A^R$$

$$R_B = R_B^L \mid R_B^R$$

6.2.3.2.3.15 The three 64-bit DES keys $K_{S1}$, $K_{S2}$ and $K_{S3}$ are calculated according the formulas:

$$K_{S1} := MAC\ (R_A^L \mid R_B^L, K_{AB}) = DES\ (K_3, DES^{-1}(K_2, DES(K_1, R_A^L \mid R_B^L)))$$

$$K_{S2} := MAC\ (R_A^R \mid R_B^R, K_{AB}) = DES\ (K_3, DES^{-1}(K_2, DES(K_1, R_A^R \mid R_B^R)))$$

$$K_{S3} := MAC\ (R_A^L \mid R_B^L, K'_{AB}) = DES\ (K_1, DES^{-1}(K_2, DES(K_3, R_A^L \mid R_B^L)))$$

where **|** is the concatenation operator, *DES* is the DES encryption function, and $DES^{-1}$ is the inverse DES encryption function, or decryption.

6.2.3.2.3.16 The length of a level 1 triple key is equal to 192 bits including parity bits. In order to get three 64-bit DES-keys for the single DES with modified MAC algorithm 3 from the three 64-bit session key generator outputs, each eighth bit of the 192 bits should be set to an odd-parity value as defined in the standard [ANSI X3.92]. However, setting the parity bits is an implementation matter where the key is internal to an equipment.

### 6.2.3.3 Communication procedures

#### 6.2.3.3.1 Connection establishment

6.2.3.3.1.1 The following procedures are applied during connection establishment:

- The safety address information is passed to the CFM
- The peer entity authentication procedure is applied.

#### 6.2.3.3.2 Data transfer

© This document has been developed and released by UNISIG

6.2.3.3.2.1 The purpose of the data transfer phase is to permit the safe transfer of normal user data between the two SaS-users connected by the safety connection. The following procedures are applied:

- The message origin authentication procedure (refer to section 6.2.3.2.1.1) for normal data;
- The service primitive's procedures provided by the transport layer.

#### 6.2.3.3.3 Connection release

6.2.3.3.3.1 The safety connection is released by a SaS-user request, by a transport service provider action, or by an error handling action of the safety layer.

*6.2.3.3.3.2* The authentication of the connection release phase is not required.

#### 6.2.3.3.4 Error handling

6.2.3.3.4.1 Errors can occur during the connection set-up in the peer entity authentication, during the data transfer, and in the management of the safety protocol.

6.2.3.3.4.2 All errors have to be reported to the local SaS-user by the Sa-REPORT.indication or by the Sa-DISCONNECT.indication primitives.

6.2.3.3.4.3 Different error cases are handled by different strategies:

- Ignore the safety relevant event;
- Optionally, ignore the safety relevant event and indicate the error to the SaS-user by Sa-REPORT.indication primitive;
- Release the safety connection, release of transport connection and indicate the error to the SaS-user by Sa-DISCONNECT.indication primitive.

6.2.3.3.4.4 It is the matter of the SaS user to react to the indicated event in a proper way.

6.2.3.3.4.5 Note: Registration of safety relevant errors is the matter of the application.

### 6.2.4 Time sequences

6.2.4.1 The flow of control information and user data is described in this chapter.

6.2.4.2 **Connection establishment**

6.2.4.2.1 When the Sa-CONNECT.request primitive requests a safety connection, the safety layer requests transport connection establishment by means of the service primitive T-CONNECT.request. This service primitive includes the first message of the peer entity authentication procedure (AU1 SaPDU) as user-data.

6.2.4.2.2 Note: AU1 and AU2 SaPDUs are exchanged by means of T-CONNECT primitives.

6.2.4.2.3 The called peer transport entity indicates the transport connection establishment request to its safety layer using the service primitive T-CONNECT.indication. The AU1 SaPDU is forwarded to the safety layer in this service primitive as user-data. At the end of the first step the called safety layer entity evaluates the AU1 SaPDU.

*© This document has been developed and released by UNISIG*

6.2.4.2.4 If it is accepted, the safety entity responds to the TC establishment request by means of the service primitive T-CONNECT.response. It includes the second message of the peer entity authentication protocol (AU2 SaPDU) as user-data.

6.2.4.2.5 There is no QoS negotiation between peer entities.

6.2.4.2.6 AU1 and AU2 SaPDUs can be used for safety feature negotiation, corresponding to a version number. The initiating safety entity may request in the AU1 SaPDU a certain safety feature. The safety feature in the AU2 SaPDU will be the version accepted by the responding safety entity. If the initiating safety entity requests a safety feature not available, the safety feature in the AU2 SaPDU will be the default value.

6.2.4.2.7 On reception, the calling transport entity informs the safety layer of the successful establishment of the transport connection using the service primitive T-CONNECT.confirmation. The AU2 SaPDU is forwarded to the safety layer as user-data within this service primitive.

6.2.4.2.8 The safety entity then generates the AU3 SaPDU that contains the third message of the authentication protocol (auth3), as user-data. It uses the T-DATA.request service primitive to forward this message to the transport layer.

6.2.4.2.9 On reception, the transport entity uses the service primitive T-DATA.indication to forward the AU3 SaPDU to the safety layer as user-data. The safety entity evaluates the AU3 SaPDU.

6.2.4.2.10 In the case of a successful AU3 SaPDU evaluation, the safety entity forwards the service primitive Sa-CONNECT.indication to the safety user (i.e. ATP application).

6.2.4.2.11 If the safety user accepts the safety connection establishment request, it responds using the service primitive Sa-CONNECT.response.

6.2.4.2.12 The safety entity on the called side sends the authentication response message in the AR SaPDU by means of the T-DATA.request and T-DATA.indication primitives to its peer safety entity.

6.2.4.2.13 Note: The authentication response message is not required by the peer entity authentication procedure. It is added to provide an OSI-like confirmed service.

6.2.4.2.14 After a successful evaluation of this SaPDU including the authentication data, the safety entity informs the SaS-user that a safety connection is now successfully established, using the service primitive Sa-CONNECT.confirmation.

6.2.4.2.15 When the Sa-CONNECT.confirmation is received, the calling SaS user is able to send data to the peer user through the safe connection. The called SaS user is able to request the data transfer immediately after the Sa-CONNECT.response primitive.

*© This document has been developed and released by UNISIG*

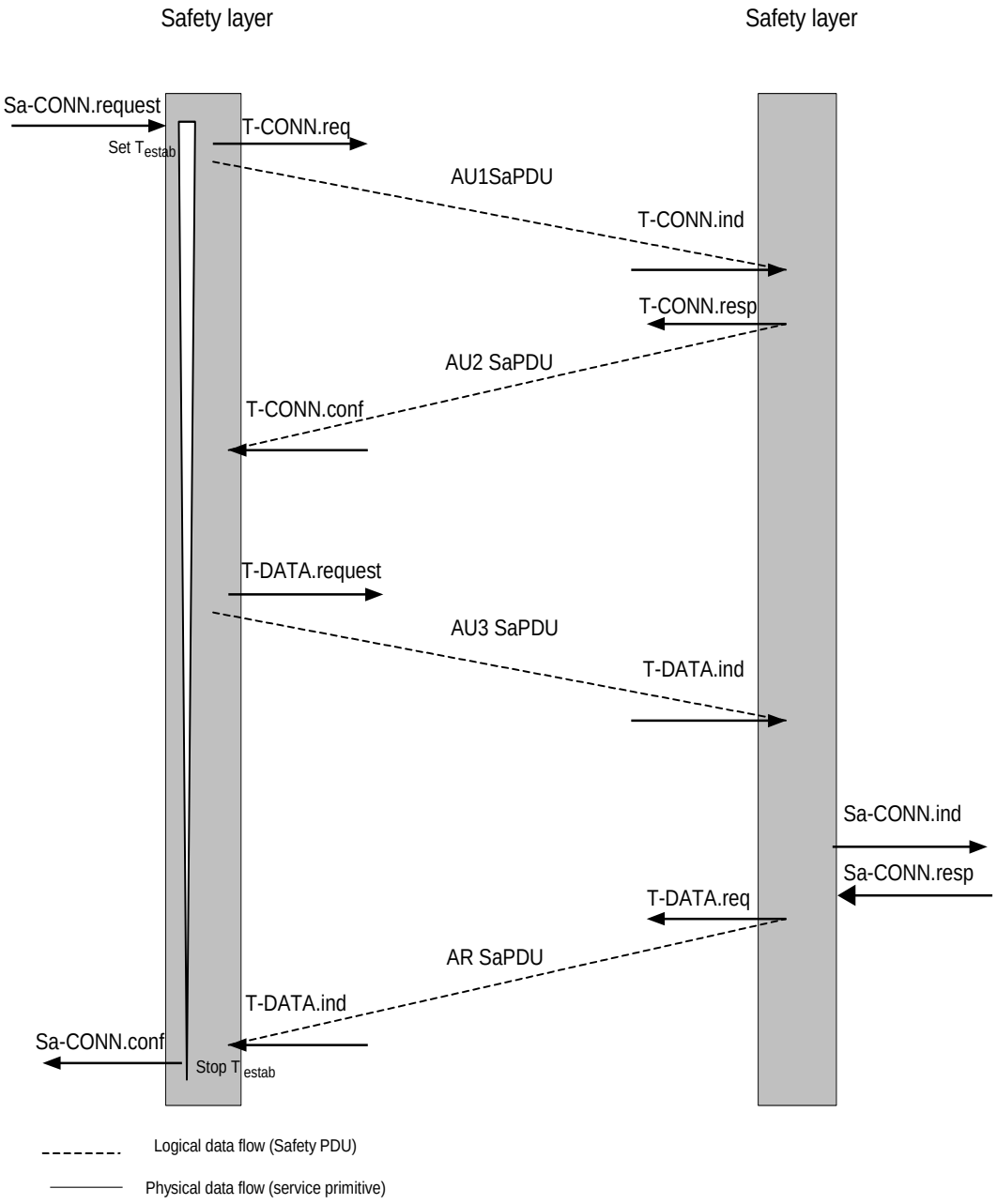



**Figure 7 Time sequence during connection establishment**

© This document has been developed and released by UNISIG

6.2.4.2.16 The maximum connection establishment delay timer is used for detecting unacceptable delay during the connection establishment. The timer $T_{estab}$ is set after reception of the Sa-CONNECT.request and is stopped before generation of the Sa-CONNECT.confirmation. In the case of time-out, a Sa-DISCONNECT.indication is generated including a proper reason. All SaPDUs will be ignored if received after the timer elapses.

6.2.4.2.17 The safety layer entity of an RBC must be able to handle the establishment of more than one safe connection at the same time. The on-board system must be able to have contact with two entities at the same time to allow seamless area change. Other situations may also require this feature.

### 6.2.4.3 Data Transfer

6.2.4.3.1 The protocol sequence of Figure 8 shows how data are transmitted by the SFM. The user data of a Sa-DATA.request primitive are included in the user data part of the DT SaPDU. The transfer of the DT SaPDU uses the transport service primitives T-DATA.request and T-DATA.indication.

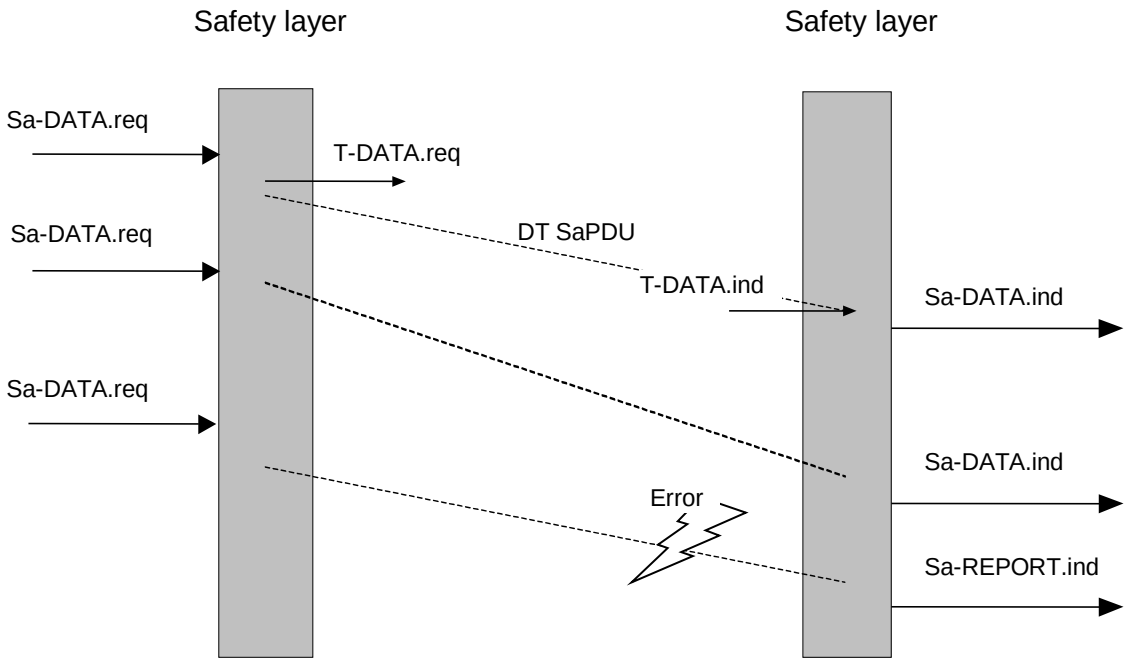



**Figure 8 Time sequence during data transfer (example)**

6.2.4.3.2 The receiving safety layer entity:

- Checks the format of the SaPDU and the protocol control information;
- Checks the MAC and integrity.

6.2.4.3.3 The user data of a safe transmitted DT SaPDU are included in a Sa-DATA.indication primitive.

6.2.4.3.4 In the case of a safety problem with the DT SaPDU, the Sa-REPORT.indication or the Sa-DISCONNECT.indication indicates this to the safety user.

### 6.2.4.4 Connection Release

© This document has been developed and released by UNISIG

6.2.4.4.1 The connection release is requested by means of the primitive Sa-DISCONNECT.request. The safety layer then requests the transport layer to disconnect by means of T-DISCONNECT.request. The DI SaPDU is included in the user data of the T-DISCONNECT.request primitive (Figure 9).

6.2.4.4.2 Peer entities are informed about the disconnection by means of T-DISCONNECT.indication and Sa-DISCONNECT.indication.

6.2.4.4.3 Authentication of the connection release phase is not required.

6.2.4.4.4 In the case of a service provider or safety layer originated connection release, this release will be indicated to both SaS-users by Sa-DISCONNECT.indication containing the respective reason.

6.2.4.4.5 Note: In the case of a service-provider-caused release, SaPDUs can be lost due to corrupted TPDUs.

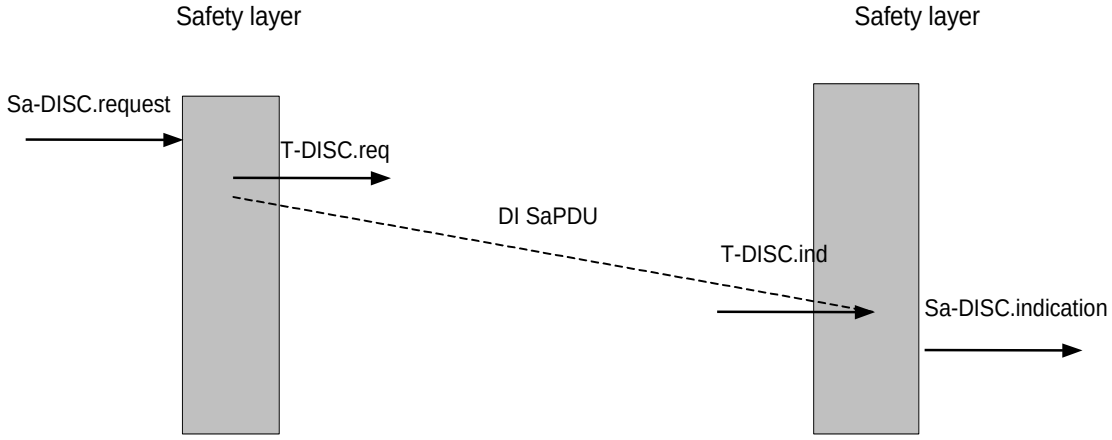



**Figure 9 Time sequence during connection release (SaS-user originated)**

## 6.2.5 Structure and encoding of safety PDUs

### 6.2.5.1 General structure of SaPDUs

6.2.5.1.1 All the safety protocol data units (SaPDUs) shall contain an integral number of octets. The octets in a SaPDU are numbered starting from 1 and increasing in the order they are put into a SaPDU. The bits in an octet are numbered from 8 to 1, where bit 1 is the lowest order bit. If a SaPDU field uses more than one octet, bit 8 of the first octet contains the most significant bit of the field.

6.2.5.1.2 When consecutive octets are used to represent a binary number, the lower octet number has the most significant value.

6.2.5.1.3 The meaning of an indication ”Reserved” is:

- The transmitting side has to insert the value ”0”;
- The receiving side has to interpret as ”Don’t care”.

6.2.5.1.4 SaPDUs shall contain, in the following order:

© This document has been developed and released by UNISIG

- The header (consisting of the message type identifier field and the direction flag field);
- The data field (if present);
- The MAC field (if applicable).

6.2.5.1.5 The structure is illustrated in Table 6.

**Table 6 Structure of a Safety PDU**

| **Header**<br>**Type + Direction** | **Data** | **MAC**<br>Not used for AU1 or DI SaPDU |
|---|---|---|
| 1 Octet | Variable | 8 octets |

#### 6.2.5.1.6 Message Type Identifier field

6.2.5.1.6.1 The message type identifier (MTI) specifies the type of the SaPDU (Table 7).

**Table 7 Safety PDUs**

| **Type** | **Type Code** | **Name** |
|---|---|---|
| AU1 SaPDU | 0001 | First authentication SaPDU (AU1) |
| AU2 SaPDU | 0010 | Second authentication SaPDU (AU2) |
| AU3 SaPDU | 0011 | Third authentication SaPDU (AU3) |
| AR SaPDU | 1001 | Response to third authentication SaPDU (AR) |
| DT SaPDU | 0101 | Data SaPDU (DT) |
| DI SaPDU | 1000 | Disconnect SaPDU (DI) |

#### 6.2.5.1.7 Direction flag field

6.2.5.1.7.1 The direction flag is used as a protection against reflection attacks. It is initialised during connection set-up. Its value is zero when the connection initiator transmits a message and one when the responder of the connection transmits a message.

6.2.5.1.7.2 The message type identifier field and direction flag field together make up the header.

#### 6.2.5.1.8 MAC field

6.2.5.1.8.1 The MAC computation is specified in the section 6.2.2.

### 6.2.5.2 Connection establishment PDU

6.2.5.2.1 AU1 and AU2 SaPDUs are exchanged by means of T-CONNECT primitives.

6.2.5.2.2 The **first authentication SaPDU** consists of the fields specified in Table 8.

**Table 8 Structure of the AU1 SaPDU**

| **Octet** | **Bit**<br>**8765 4321** | **Field name** | **Field** |
|---|---|---|---|
| 1 | xxx. ....<br>000. ....<br>001. ....<br>010. ....<br>011. .... | ”ETY” | ETCS ID type of the field ”SA”<br>Radio in-fill unit<br>RBC<br>Engine<br>Reserved for Balise |

© This document has been developed and released by UNISIG

| | | | |
|---|---|---|---|
| | 100. ....<br>101. ....<br>110. .... | | Reserved for Field element (level crossing etc)<br>Key management entity<br>Interlocking related entity |
| 1 | ...0 001. | ”MTI” | Message Type Identifier: **AU1** |
| 1 | .... ...0 | ”DF” | Direction Flag: ’0’B indicates the direction to the responder |
| 2<br>3<br>4 | xxxx xxxx<br>xxxx xxxx<br>xxxx xxxx | ”SA” | Calling ETCS ID |
| 5 | Xxxx xxxx<br>0000 0001 | ”SaF” | Requested Safety feature<br>Single DES with modified MAC algorithm 3<br>All other values are reserved |
| 6<br>...<br>13 | Xxxx xxxx<br>...<br>xxxx xxxx | "$R_B$" | Random number $R_B$ of the first authentication message |

6.2.5.2.3 The **second authentication SaPDU** consists of the fields specified in Table 9.

**Table 9 Structure of the AU2 SaPDU**

| Octet | Bit<br>8765 4321 | Field name | Field |
|---|---|---|---|
| 1 | xxx. .... | ”ETY” | ETCS ID type of the field ”SA”<br>See Table 8 |
| 1 | ...0 010. | ”MTI” | Message Type Identifier: **AU2** |
| 1 | .... ...1 | ”DF” | Direction Flag: ’1’B indicates the direction to the initiator |
| 2<br>3<br>4 | xxxx xxxx<br>xxxx xxxx<br>xxxx xxxx | ”SA” | Responding ETCS Id. |
| 5 | xxxx xxxx<br>0000 0001 | ”SaF” | Accepted safety features.<br>Single DES with modified MAC algorithm 3<br>All other values are reserved. |
| 6<br>...<br>13 | xxxx xxxx<br>...<br>xxxx xxxx | "$R_A$" | Random number $R_A$ of the second authentication message |
| 14<br>...<br>21 | xxxx xxxx<br>...<br>xxxx xxxx | | MAC field.<br>The MAC is computed according to the rules given in the peer entity and message origin authentication procedure. |

6.2.5.2.4 The **third authentication SaPDU** consists of the fields specified in Table 10.

**Table 10 Structure of the AU3 SaPDU**

| Octet | Bit<br>8765 4321 | Field name | Field |
|---|---|---|---|
| 1 | 000. .... | ”ETY” | Reserved |
| 1 | ...0 011. | ”MTI” | Message Type Identifier: **AU3** |
| 1 | .... ...0 | ”DF” | Direction Flag: ’0’B indicates the direction to the responder |
| 2<br>...<br>9 | xxxx xxxx<br>...<br>xxxx xxxx | | MAC field.<br>The MAC is computed according to the rules given in the peer entity and message origin authentication procedure |

*© This document has been developed and released by UNISIG*

6.2.5.2.5 The **Authentication Response SaPDU** consists of the fields specified in Table 11.

**Table 11 Structure of the AR SaPDU**

| Octet | Bit 8765 4321 | Field name | Field |
|---|---|---|---|
| 1 | 000. .... | ”ETY” | Reserved |
| 1 | ...1 001. | ”MTI” | Message Type Identifier: **AR** |
| 1 | .... ...1 | ”DF” | Direction Flag: ’1’B indicates the direction to the initiator |
| 2<br>...<br>9 | xxxx xxxx<br>...<br>xxxx xxxx | | MAC field.<br>the MAC computed according to the rules given in the peer entity and message origin authentication procedure |

### 6.2.5.3 Data Transfer SaPDU

6.2.5.3.1 The **Data SaPDU** consists of the fields specified in Table 12.

**Table 12 Structure of the DT SaPDU**

| Octet | Bit 8765 4321 | Field name | Field |
|---|---|---|---|
| 1 | 000. .... | | |
| 1 | ...0 101. | ”MTI” | Message Type Identifier: **DT** |
| 1 | .... ...x | ”DF” | Direction Flag |
| 2<br>...<br>2+n-1 | xxxx xxxx<br>...<br>xxxx xxxx | | User data (length n>=1 octet):<br>user data of the corresponding SaPDU |
| 2+n<br>...<br>2+n+7 | xxxx xxxx<br>...<br>xxxx xxxx | | MAC field. |

### 6.2.5.4 Disconnect SaPDU

6.2.5.4.1 The **Disconnect SaPDU** consists of the fields specified in Table 13.

**Table 13 Structure of the DI SaPDU**

| Octet | Bit 8765 4321 | Field name | Field |
|---|---|---|---|
| 1 | 000. .... | | |
| 1 | ...1 000. | ”MTI” | Message Type Identifier: **DI** |
| 1 | .... ...x | ”DF” | Direction flag. |
| 2 | xxxx xxxx | | Reason field: the reason for the disconnect. |
| 3 | xxxx xxxx | | SUB-reason field: the sub-reason for the disconnect. |

## 6.2.6 State table

6.2.6.1 The state transition diagram and the state table are symmetrical for on-board and trackside SFM.

*© This document has been developed and released by UNISIG*

### 6.2.6.2 General

6.2.6.2.1 This section describes the safety protocol in terms of state tables. The state tables show the state of a safety layer entity, the events that occur in the protocol, the actions taken and the resultant state. The state tables are conceptual and do not impose any constraints on the implementation.

6.2.6.2.2 The state tables also define the mapping between safety service primitives and protocol events that safety service users (SaS users) can expect.

6.2.6.2.3 The state tables do not necessarily describe all possible combinations of sequences of events at safety and transport service boundary, nor do they describe the exact mapping between SaPDUs and TSDUs.

### 6.2.6.3 Conventions

6.2.6.3.1 States are represented in the tables by their abbreviation, as defined in Table 14.

**Table 14 States**

| State abbreviation | State name |
|---|---|
| WFTC | Wait for transport connection |
| WFAR | Wait for the authentication response SaPDU |
| DATA | Safety connection is opened and ready for data transfer |
| WFAU3 | Wait for the third authentication message |
| WFRESP | Wait for Sa-CONNECT.response |
| IDLE | Safety connection is closed or does not exist |

© This document has been developed and released by UNISIG

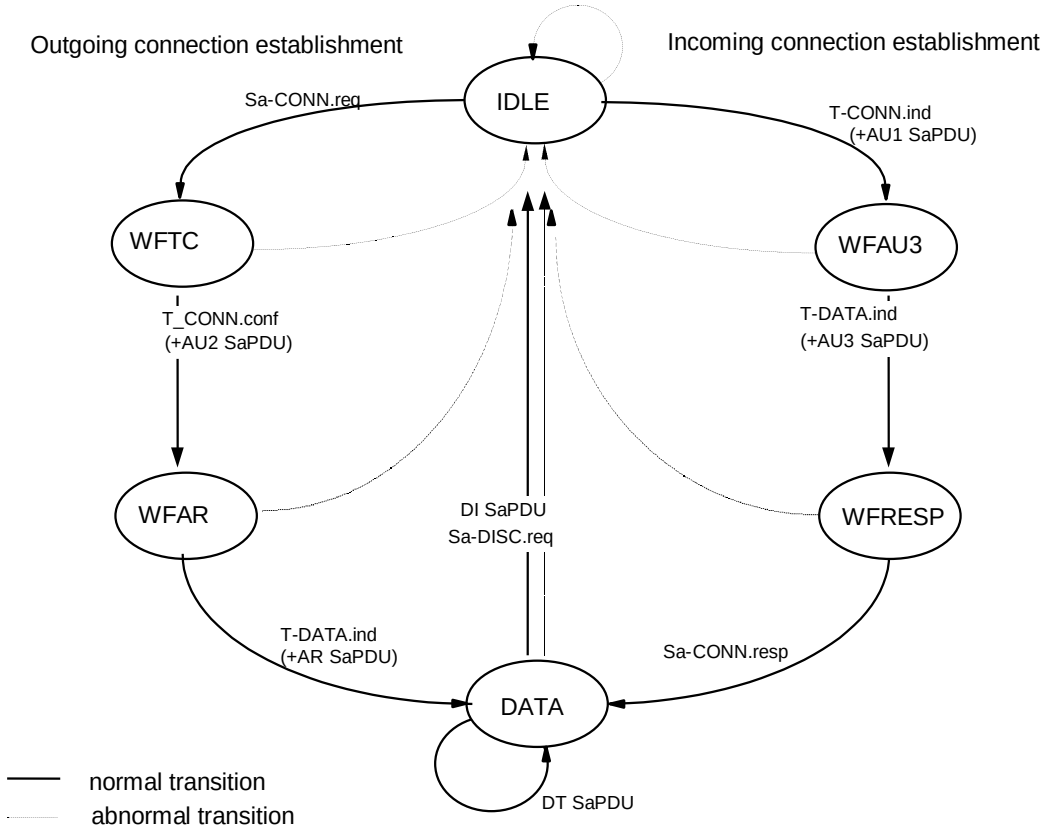



**Figure 10 State transition diagram of the safety layer entity**

6.2.6.3.2 The intersection of each state and incoming event that is invalid is left blank in the state tables. The action to be taken in this case shall be one of the following:

- for an event related to the safety service (i.e. coming from the SaS-user), take no action;
- for an event related to a received SaPDU, follow the procedure for treatment of protocol errors if the state of the supporting transport connection makes it possible;
- for an event falling into neither of the above categories (including those which are impossible by the definition of the behaviour of the safety entity or SaS-provider), take no action.

6.2.6.3.3 At each intersection of state and event which is valid, the state tables specify an action which may include one of the following:

- one action constituted of a list of any number of outgoing events (none, one, or more) given by their abbreviation defined in Table 16 followed by certain special actions (see Table 18), if applicable, and the abbreviation of the resultant state (see Table 14);
- conditional actions separated by a semi-colon (;). Each conditional action contains a predicate followed by a colon (:) and by an action as defined in a). The predicates are Boolean expressions given by their abbreviation and defined in Table 17. Only the action corresponding to the true predicate shall be taken.

© This document has been developed and released by UNISIG

6.2.6.3.4 There is a unique association between the safety connection and the transport connection used. The mapping of the local references (SaCEPID and TCEPID) is a matter of the implementation.

6.2.6.3.5 Table 15 specifies the names and abbreviation of the incoming events classified as event originated by TS-provider, SaS-user or safety layer entity.

**Table 15 Incoming events**

| Abbreviation | Origin of event | Name |
|---|---|---|
| Sa-CONN.req | SaS-user | Sa-CONNECT.request primitive |
| Sa-CONN.resp | SaS-user | Sa-CONNECT.response primitive |
| Sa-DATA.req | SaS-user | Sa-DATA.request primitive |
| Sa-DISC.req | SaS-user | Sa-DISCONNECT.request primitive |
| T-DISC.ind | TS-provider | T-DISCONNECT.indication primitive |
| T-CONN.ind (+AU1SaPDU) | TS-provider | T-CONNECT.indication primitive |
| T-CONN.conf (+AU2SaPDU) | TS-provider | T-CONNECT.confirmation primitive |
| AU3 SaPDU | Safety layer entity | Authentication 3 SaPDU |
| AR SaPDU | Safety layer entity | Authentication response SaPDU |
| DI SaPDU | Safety layer entity | Disconnect Request SaPDU |
| DT SaPDU | Safety layer entity | Data SaPDU |
| time-out Testab | Safety layer entity | Connection establishment timer |

6.2.6.3.6 Table 16 specifies the names and abbreviations of the outgoing events classified as event originated by SaS-provider, TS-user or safety layer entity.

**Table 16 Outgoing events**

| Abbreviation | Origin of event | Name |
|---|---|---|
| Sa-CONN.ind | SaS-provider | Sa-CONNECT.indication primitive |
| Sa-CONN.conf | SaS-provider | Sa-CONNECT.confirmationprimitive |
| Sa-DATA.ind | SaS-provider | Sa-DATA.indication primitive |
| Sa-DISC.ind | SaS-provider | Sa-DISCONNECT.indication primitive |
| Sa-REPORT.ind | SaS-provider | Sa-REPORT.indication primitive |
| T-CONN.req (+AU1SaPDU) | TS-user | T-CONNECT.request primitive |
| T-CONN.resp (+AU2SaPDU) | TS-user | T-CONNECT.response primitive |
| T-DISC.req | TS-user | T-DISCONNECT.request primitive |
| AU3 SaPDU | Safety layer entity | Authentication 3 SaPDU |
| AR SaPDU | Safety layer entity | Authentication response SaPDU |

*© This document has been developed and released by UNISIG*

| Abbreviation | Origin of event | Name |
|---|---|---|
| DI SaPDU | Safety layer entity | Disconnect Request SaPDU |
| DT SaPDU | Safety layer entity | Data SaPDU |

**Table 17 Predicates**

| Name | Description |
|---|---|
| Pre0 | Sa-CONNECT. request unacceptable<br>• at least the following parameter is required : application type<br>• application type error |
| Pre1 | Unacceptable T-CONNECT.indication,<br>• at least the following parameters are required : application type, user data<br>• application type error<br>Unacceptable AU1 SaPDU<br>• AU1 SaPDU format error<br>• ETY,MTI,DF or SaF error<br>• KMAC not available |
| Pre2 | Unacceptable T-CONNECT.confirmation,<br>• at least the following parameter is required: user data<br>Unacceptable AU2 SaPDU<br>• AU2 SaPDU format error<br>  • ETY,MTI,DF or SaF error<br>  • KMAC not available<br>  • MAC error |
| Pre3 | Unacceptable AU3 SaPDU<br>• AU3 SaPDU format error<br>  • ETY, MTI or DF error<br>  • MAC error |
| Pre4 | Unacceptable AR SaPDU<br>• AR SaPDU format error<br>  • ETY, MTI or DF error<br>• MAC error |
| Pre 5 | Erroneous SaPDU<br>• MAC error of DT SaPDU |
| Pre6 | Unacceptable DT SaPDU<br>• DT SaPDU length error<br>• MTI error<br>• DF error (condition: no MAC error) |

6.2.6.3.7 The state table specifies the precise protocol to provide interoperability, but does not specify the implementation of the protocol.

**Table 18 Timer definitions**

| Symbol | Name | Definition |
|---|---|---|
| $T_{estab}$ | Connection establishment time | An upper bound for the time after which the local safety entity will initiate the error handling procedure, if it does not receive the authentication response message. |

**Table 19 Integrity actions**

© This document has been developed and released by UNISIG

| Abbreviation | Action |
|---|---|
| a5 | Set timer $T_{estab}$ |
| a6 | Stop timer $T_{estab}$ |
| a19 | Stop all timers; reset all counters. |

**Table 20 State table**

| State<br>Event | IDLE | WFTC | WFAR | DATA | WFAU3 | WFRESP |
|---|---|---|---|---|---|---|
| Sa-CONN.req | Pre0: Sa-DISC.ind, IDLE;<br>not Pre0: T-CONN.req (AU1 SaPDU), a5, WFTC | | | | | |
| Sa-CONN.resp | | | | | | AR SaPDU, DATA |
| Sa-DATA.req | | | | DT SaPDU, DATA | | |
| Sa-DISC.req | | T-DISC.req a19, IDLE Note1 | T-DISC.req (+DI SaPDU), a19, IDLE | T-DISC.req (+DI SaPDU), a19, IDLE | | T-DISC.req (+DI SaPDU) a19, IDLE |
| T-CONN.ind (+AU1SaPDU) | Pre1:T-DISC.req (+DI SaPDU), IDLE;<br>not Pre1: T-CONN.resp (+AU2 SaPDU) WFAU3 | | | | | |
| T-CONN.conf (+AU2SaPDU) | | Pre2: Sa-DISC.ind, T-DISC.req, a19, IDLE Note1<br>not Pre2: AU3 SaPDU, WFAR | | | | |
| T-DISC.ind<br>or<br>T-DISC.ind (+DI SaPDU) | | Sa-DISC.ind, a19, IDLE | Sa-DISC.ind, a19, IDLE | Sa-DISC.ind a19, IDLE | a19, IDLE | Sa-DISC.ind a19, IDLE |
| AU3 SaPDU | | | | | Pre3:<br>T-DISC.req (+DI SaPDU), a19, IDLE<br>not Pre3: Sa-CONN.ind, WFRESP | |

© This document has been developed and released by UNISIG

| State<br>Event | IDLE | WFTC | WFAR | DATA | WFAU3 | WFRESP |
|---|---|---|---|---|---|---|
| AR SaPDU | | | not Pre4: Sa-CONN.conf, a6, DATA;<br>Pre4: Sa-DISC.ind, T-DISC.req (+DI SaPDU), a19, IDLE | | | |
| DT SaPDU | | | | not Pre5 and not Pre6: Sa-DATA.ind, DATA;<br>Pre5: Sa-REPORT.ind, DATA Note2<br>Pre6: Sa-DISC.ind, T-DISC.req (+DI SaPDU), a19, IDLE | | |
| time-out $T_{estab}$ | | Sa-DISC.ind, T-DISC.req, a19, IDLE Note1 | Sa-DISC.ind, T-DISC.req (+DI SaPDU), a19, IDLE | | | |
| Notes:<br>1. The DI SaPDU is not contained.<br>2. Optional Sa-REPORT.indication delivered to the SaS user, if supported. | | | | | | |

## 6.3 Safety Protocol Management

### 6.3.1 Functions of the Safety Protocol Management

6.3.1.1 The safety protocol management defines the configuration management needed to handle the parameters of the safety protocol, and the supervision and diagnostics of the safety protocol. The main emphasis is placed on achieving technical interoperability between the on-board and the trackside unit with respect to the safety protocol management.

6.3.1.2 All details of the specification, which are implementation dependent like the generation, storage, and deletion of keys, or error logging are not covered by this specification.

6.3.1.3 The over-the-air updating of keys is possible, but not specified here, see [Subset-137].

6.3.1.4 The management of the safety layer protocol is embedded in the SFM sub-system. Parts of it are clearly safety related and have to be realised in a safe environment whereas other parts are not. The details depend on the particular implementation and are not covered by this specification.

### 6.3.2 Configuration Management

6.3.2.1 The configuration management defines the parameters needed for the execution of the safety protocol and its management, and the functions to manage them.

6.3.2.2 **Address Parameters**

© This document has been developed and released by UNISIG

6.3.2.2.1 The safety protocol uses the ETCS Identities for addressing. The ETCS Identities are unique within the scope of the respective ETCS ID type. The ETCS ID together with the application type identifies the safety service user.

**Table 21 ETCS Identity (see ETCS SRS [Subset-026] chapter 7)**

| ETCS ID | Range of values<br>**Octet1 Octet2 Octet3**<br>**8765 4321 8765 4321 8765 4321** | Description |
|---|---|---|
| ETCS ID of on-board | tttt tttt tttt tttt tttt tttt | |
| ETCS ID of RBC | cccc cccc ccrr rrrr rrrr rrrr<br><br>.... .... ..11 1111 1111 1111 | c…c Country or region ID<br>r…r RBC ID<br>ETCS ID unknown |
| ETCS ID of radio in-fill unit (ETCS level1) | cccc cccc ccrr rrrr rrrr rrrr<br><br>.... .... ..11 1111 1111 1111 | c…c Country or region ID<br>r…r RIU ID<br>ETCS ID unknown |
| ETCS ID of KM entity | cccc cccc cckk kkkk kkkk kkkk | c…c Country or region ID<br>k…k Key management entity ID |

6.3.2.2.2 Note: The definition of ETCS ID structure and values is out of scope for this FIS.

6.3.2.2.3 Identities are used during the connection set-up to compute the corresponding safety association, i.e. the ETCS IDs are relevant for the execution of the safety procedure peer entity authentication.

6.3.2.2.4 A safety association is defined between two ETCS-Identities as soon as they share a common authentication key to set up a safe connection. Besides the authentication key, also the other parameters have to be defined for every safety association.

6.3.2.2.5 Additionally, the transport service access points (TSAPs) are used by the safety layer to access the transport layer.

#### 6.3.2.3 Timer Parameter

6.3.2.3.1 The parameter maximum connection establishment delay is used for detecting unacceptable delay during the connection establishment.

**Table 22 Safety layer timer parameter**

| Parameter | Symbol | Applied value | Comments |
|---|---|---|---|
| Maximum connection establishment delay | $T_{estab}$ | 40 s | Depends on the communication network |

### 6.3.3 Supervision and Diagnostics

6.3.3.1 The supervision and diagnostics describes the error management of the safety layer and the monitoring and auditing of safety relevant events.

6.3.3.2 The error management defines the error handling, and the error reporting to the application layer, as far as it is needed for interoperability reasons.

6.3.3.3 Note: Error logging by SFM is not required. It has to be done by the application, if required.

© This document has been developed and released by UNISIG

#### 6.3.3.4 Error Reporting

6.3.3.4.1 All safety relevant errors that occur in the safety layer which are treated by the application have to be reported to the application immediately after their occurrence. Errors handled internally by the safety layer management, may be reported to the application but do not have to be. There are two possibilities for reporting errors to the application:

- If the error leads to a mandatory connection release, it can be reported to the application using the service primitive Sa-DISCONNECT.indication. The application is informed about the type of the error using the parameter **disconnect reason**.
- If the error is only treated internally by the safety layer management or does not lead to a mandatory connection release it can be reported optionally to the application using the service primitive Sa-REPORT.indication. The application is informed about the type of the error by the parameter pair (**reason code, sub-reason code)**.

#### 6.3.3.5 Error Handling

6.3.3.5.1 If an error occurs in the safety layer the error management has to undertake the following actions depending on the reason and sub-reason of this error. One indicated reason may be caused by different sub-reasons which may be detected by symptoms requiring different error handling actions. The pairs (reason code, sub-reason code) are applied in the Sa-DISCONNECT.indication and Sa-REPORT.indication to indicate the type of the error to the user of the service.

6.3.3.5.2 An error handling action implies the sending of T-DISCONNECT.request (+DI SaPDU), if requested according to state table.

6.3.3.5.3 When error information is transmitted to the application by Sa-DISCONNECT.indication, it is the responsibility of the application for further action.

6.3.3.5.4 The error indication provided by T-DISCONNECT.indication shall be handled by the safety layer:

- When reason = Network error is received, this error is forwarded to the application.
- The reason = Called TS user not available should not be received from the Communication Layer, as the ATP is supposed to be supported by the peer entity. However, if this reason is received by the safety layer, the application will be informed.

**Table 23 Normal release**

| Reason Code | Sub-reason Code | Description | Error handling action |
|---|---|---|---|
| 0 | | Normal release requested by peer SFM user | Sa-DISCONNECT.indication |

**Table 24 Sub-reasons for the reason 'No transport service available'**

© This document has been developed and released by UNISIG

| Reason Code | Sub-reason Code | Description | Error handling action |
|---|---|---|---|
| 1 | 1 | Network error | Sa-DISCONNECT.indication<br>The application should try to establish again the connection |
| 1 | 2 | Network resource not available | Sa-DISCONNECT.indication<br>The application should try to establish again the connection |
| 1 | 3 | Service or option is temporarily not available | Sa-DISCONNECT.indication<br>The application should try to establish again the connection with a modified parameter. |
| 1 | 5 | Reason unknown | Sa-DISCONNECT.indication |
| 1 | 6 | Called TS user not available | Sa-DISCONNECT.indication<br>The application should try to establish again the connection with short dialling code |
| 1 | 8 | No Mobile Termination has been registered | Sa-DISCONNECT.indication<br>The application should re-try network registration |
| Note: 1.The sub-reason is equivalent to the reason of T-DISCONNECT.indication.<br>2. Sub-reasons are a matter of implementation. The error codes are not transmitted via the air interface | | | |

**Table 25 Sub-reasons for the reason 'Missing parameter or invalid parameter value'**

| Reason Code | Sub-reason Code | Description | Error handling action |
|---|---|---|---|
| 3 | 2 | Missing authentication key | Sa-DISCONNECT.indication. |
| 3 | 3 | Other problem related to the key management (e.g. loss of session key). | Sa-DISCONNECT.indication. The SFM user can set-up a new connection. |
| 3 | 4 | Authentication key not currently valid | Sa-DISCONNECT.indication. |
| 3 | 29 | Requested safety feature is not supported | Sa-DISCONNECT.indication |

**Table 26 Sub-reasons for the reason 'Invalid MAC'**

| Reason Code | Sub-reason Code | Description | Error handling action |
|---|---|---|---|
| 4 | 1 | MAC error | Sa-REPORT.indication |
| 4 | 2 | MAC error in AU2 SaPDU. | Sa-DISCONNECT.indication. |
| 4 | 3 | MAC error in AU3 SaPDU | T-DISCONNECT.request. |
| 4 | 4 | MAC error in AR SaPDU | Sa-DISCONNECT.indication |

**Table 27 Sub-reasons for the error type 'failure in sequence integrity'**

*© This document has been developed and released by UNISIG*

| Reason Code | Sub-reason Code | Description | Error handling action |
|---|---|---|---|
| 5 | 1 | Replay of authentication message (AU1 SaPDU, AU2 SaPDU, AU3 SaPDU, AR SaPDU) after connection establishment. Error code is used, if the error is not covered by reason code 9. | Sa-DISCONNECT.indication |

6.3.3.5.5 Error type: Failure in the direction flag

6.3.3.5.6 This check is performed after the check of the MAC (not in the case of AU1 or DI SaPDU). If there is a transmission error that affects the flag, the MAC will detect this, and the reaction will be as in Table 25. If the MAC is correct, but the flag is not correct, there will be a SA-DISCONNECT.indication.

**Table 28 Sub-reasons for the reason 'Failure in the direction flag'**

| Reason Code | Sub-reason Code | Description | Error handling action |
|---|---|---|---|
| 6 | 1 | Value of direction flag '0' instead of '1' | Sa-DISCONNECT.indication<br>The application is supposed to request a new connection establishment. |
| 6 | 2 | Value of direction flag '1' instead of '0' | Sa-DISCONNECT.indication (after previous Sa-CONNECT.indication)<br>The application is supposed to request a new connection establishment. |

.

**Table 29 Sub-reasons for the reason 'Time out at connection establishment'**

| Reason Code | Sub-reason Code | Description | Error handling action |
|---|---|---|---|
| 7 | 3 | Time out of $T_{estab}$ without receiving the AR SaPDU | Sa-DISCONNECT.indication<br>The application is supposed to request a new connection establishment. |

**Table 30 Sub-reasons for the reason 'Invalid SaPDU field'**

| Reason Code | Sub-reason Code | Description | Error handling action |
|---|---|---|---|
| 8 | 1 | Invalid information field | Rejection of SaPDU |
| 8 | 4 | Invalid responding ETCS Id in AU2, i.e. ETCS-Identity does not correspond to an acceptable ETCS ID.[4] | Sa-DISCONNECT.indication |
| 8 | 5 | Invalid AU1 SaPDU : the header indicates a AU1 SaPDU, but the rest of the Sa PDU | Rejection of SaPDU |

[4] If there is a call establishment request to an unknown RBC any one of the possible RBCs can be an expected one.

© This document has been developed and released by UNISIG

| Reason Code | Sub-reason Code | Description | Error handling action |
|---|---|---|---|
| | | does not match with the structure of an AU1 SaPDU. | |

**Table 31 Sub-reasons for the reason 'Failure in sequence of the SaPDUs during connection set-up'**

| Reason Code | Sub-reason Code | Description | Error handling action |
|---|---|---|---|
| 9 | 1 | Transmission of AU1 SaPDU but a message different from AU2 SaPDU is obtained. | Sa-DISCONNECT.indication |
| 9 | 2 | Transmission of AU2 SaPDU but a message different from AU3 SaPDU is obtained. | T-DISCONNECT.request |
| 9 | 3 | Transmission of AU3 SaPDU but a message different from AR SAPDU is obtained. | Sa-DISCONNECT.indication |

**Table 32 Sub-reasons for the reason ' SaPDU length error '**

| Reason Code | Sub-reason Code | Description | Error handling action |
|---|---|---|---|
| 10 | 1 | AU1 SaPDU length error | Rejection of AU1 SaPDU |
| 10 | 2 | AU2 SaPDU length error | Sa-DISCONNECT.indication |
| 10 | 3 | AU3 SaPDU length error | T-DISCONNECT.request |
| 10 | 5 | DT SaPDU length error | Sa-DISCONNECT.indication |
| 10 | 8 | AR SaPDU length error | Sa-DISCONNECT.indication |

6.3.3.5.7 The code 127 (unknown) has to be used, when:

- no proper reason code or sub-reason code can be selected;
- the reason code or sub-reason code is undefined.

6.3.3.5.8 The reason codes 12-126 are reserved for future use. The reason codes 128-255 are reserved for national use / implementation-specific use. For these reason codes the sub-reason codes (0...126, 128...255) are also reserved for national use / implementation-specific use.

© This document has been developed and released by UNISIG

## ANNEX A. (normative) Assumptions placed on the ATP application

This section defines the conditions and constraints, which shall be covered by the ATP application when using the services provided by SFM.

a) Safety protection against occurrence of message delay, wrongly sequenced messages, message deletion and message replay shall be provided by the application, if required.

b) Safe connection monitoring should be provided, if required.

c) Service primitives have to be issued according to the sequence defined.

d) In the case of RBC area change or entrance into RBC area, the connection establishment request has to be requested as soon as possible. Normally, safe connection establishment delay is less than the value $T_{estab}$ = 40s.

e) In the case of registration with a mobile network (roaming into another GSM-R/GPRS), an additional delay has to be taken into account (refer to [Subset-093]).

f) The maximum length of an application message to be transferred is restricted to 1023 octets.

g) The transfer of application data has to be finished for both directions before a connection release is requested.

h) In the case of network caused release of the safe connection or rejected connection establishment request, the application has to request the re-establishment of the safe connection. The on-board ATP shall initiate the safe connection re-establishment. Due to possible loss of user data a re-synchronisation of the application data can be required.

i) If required, the application has to pad the user data to octet boundaries.

j) The application should check if the called ETCS ID of Sa-CONNECT.indication primitive is the same as its own ETCS ID (see fig.9).

k) The OBU application has to provide the Mobile Network ID for a safe connection request.

*© This document has been developed and released by UNISIG*

# ANNEX B. (Informative) CBC-MAC Calculation

B.1.1.1 Assume a message *m* (21 octets) with the following structure in hex notation:

```
00 01 02 03 04 05 06 07
08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 .. .. ..
```

B.1.1.2 Because it is not a multiple of 64 bits, *m* must be padded with zero bits before MAC calculation as follows:

```
00 01 02 03 04 05 06 07
08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 00 00 00
```

B.1.1.3 A 192 bit triple key is required for MAC calculation, consisting of three 64-bit DES keys (K1, K2, K3). Although not used by the DES algorithm, the key should be as defined by [ANSI], where each eighth bit (the LSB of each octet) is defined as an odd-parity bit.

B.1.1.4 In practice, the triple key to be used to calculate a MAC is the Session Key KsMAC, derived during session establishment (AU1 and AU2) from the KMAC. This example assumes that KsMAC has been generated, so the DES keys referred to below are already parts of the session key.

B.1.1.5 The first DES key (K1, bits b0 to b63 of KsMAC) is:

```
                MSB                LSB  hex
b0  - b7 : 0 0 0 0  0 0 0  1    01
b8  - b15: 0 0 0 0  0 0 1  0    02
b16 - b23: 0 0 0 0  0 1 0  0    04
b24 - b31: 0 0 0 0  0 1 1  1    07
b32 - b39: 0 0 0 0  1 0 0  0    08
b40 - b47: 0 0 0 0  1 0 1  1    0B
b48 - b55: 0 0 0 0  1 1 0  1    0D
b56 - b63: 0 0 0 0  1 1 1  0    0E
```

B.1.1.6 The structure of the DES key is defined as follows, with the greatest-weight bit being b0, b8, b16 ..., and each parity bit being b7, b15, b23 (where '|' is the concatenation operator).

```
b0       b7  b8
v        v   v
0000 0001 | 0000 0010 | 0000 0100 | 0000 0111 | 0000 1000 |
                                    0000 1011 | 0000 1101 | 0000 1110
                                                                    ^
                                                                   b63
```

or in hex notation: K1 = `01 | 02 | 04 | 07 | 08 | 0B | 0D | 0E`

© This document has been developed and released by UNISIG

B.1.1.7 The second DES key (K2, bits b64 to b127 of KsMAC) is:

```
MSB           LSB   hex
0 0 0 1   0 0 0  0   1 0
0 0 0 1   0 0 1  1   1 3
0 0 0 1   0 1 0  1   1 5
0 0 0 1   0 1 1  0   1 6
0 0 0 1   1 0 0  1   1 9
0 0 0 1   1 0 1  0   1 A
0 0 0 1   1 1 0  0   1 C
0 0 0 1   1 1 1  1   1 F
```

B.1.1.8 The third DES key (K3, bits b128 to b191 of KsMAC) is:

```
MSB           LSB   hex
0 0 1 0   0 0 0  0   2 0
0 0 1 0   0 0 1  1   2 3
0 0 1 0   0 1 0  1   2 5
0 0 1 0   0 1 1  0   2 6
0 0 1 0   1 0 0  1   2 9
0 0 1 0   1 0 1  0   2 A
0 0 1 0   1 1 0  0   2 C
0 0 1 0   1 1 1  1   2 F
```

B.1.1.9 The triple key KsMAC, consisting of the three DES keys K1 | K2 | K3, is therefore:

```
01 02 04 07 08 0B 0D 0E | 10 13 15 16 19 1A 1C 1F | 20 23 25 26 29 2A 2C 2F
```

B.1.1.10 To calculate a CBC-MAC for message *m*:

1. The DEA input register is initialised with the first 8 octets of the message, and the first DES key is used to encrypt and produce 8 octets of ciphertext output.

```
message block 1:      00 01 02 03 04 05 06 07

DES key K1:           01 02 04 07 08 0B 0D 0E

> ciphertext1:        0C 61 B5 50 4B 5C FC 5C
```

[Note that since a message block XOR'd with an initialisation vector of 0 is unchanged, it is an implementation matter whether it is done or not.]

2. Ciphertext1 is then exclusive-or'd with message block 2:

```
message block 2:      08 09 0A 0B 0C 0D 0E 0F

ciphertext1:          0C 61 B5 50 4B 5C FC 5C

> XOR2:               04 68 BF 5B 47 51 F2 53
```

3. XOR2 is now the next input to the DES algorithm, encrypting again with DES key K1:

```
XOR2:                 04 68 BF 5B 47 51 F2 53

DES key K1:           01 02 04 07 08 0B 0D 0E

> ciphertext2:        E0 13 56 59 5B 86 75 31
```

© This document has been developed and released by UNISIG

4. The process is repeated for the last message block: ciphertext2 is exclusive-or'd with message block 3 (containing the padding):

```
message block 3:      10 11 12 13 14 00 00 00

ciphertext2:          E0 13 56 59 5B 86 75 31

> XOR3:               F0 02 44 4A 4F 86 75 31
```

5. XOR3 is now the next input to the DES algorithm, again encrypting with DES key K1:

```
XOR3:                 F0 02 44 4A 4F 86 75 31

DES key K1:           01 02 04 07 08 0B 0D 0E

> ciphertext3:        DF 5E BC 63 95 68 0A 93
```

6. So far, the process has been normal single DES. Now it must be processed with modified MAC algorithm 3, that is, ciphertext3 is decrypted with DES key K2:

```
ciphertext3:          DF 5E BC 63 95 68 0A 93

DES key K2:           10 13 15 16 19 1A 1C 1F

> ciphertext4:        A1 3B 20 90 B5 D5 3D F0
```

7. Then encrypted with DES key K3:

```
ciphertext4:          A1 3B 20 90 B5 D5 3D F0

DES key K3:           20 23 25 26 29 2A 2C 2F

> CBC-MAC:            36 1D 43 1E D3 96 C1 75
```

B.1.1.11 The resulting output is the required 8-octet CBC-MAC of message *m*. Note that the message is not changed by the above process, i.e., the padding is added only for the MAC calculation and is not transmitted.

Note also that this example is generic, i.e., it excludes the process where transmitter and receiver add the destination ETCS identity (*DA*) and length of *DA*|*m* for the MAC calculation, but remove them before use, as described above in 6.2.2.9.

© This document has been developed and released by UNISIG