U-N-I-S-I-G

ERTMS/ETCS	

KMC-ETCS Entity Off-line KM FIS

REF : SUBSET-114

ISSUE: 4.0.0

DATE : 05-07-23

Company	Technical Approval	Management approval
ALSTOM		
ANSALDO		
AZD		
BOMBARDIER		
CAF		
SIEMENS		
THALES		

1. MODIFICATION HISTORY

Issue Number Date	Section Number	Modification / Description	Author
0.0.21 05-11-10	All	Draft for EEIG KMS WG review	KM WG
0.1.0 07-12-11	All	Updated as per agreements in joint meeting EEIG KMS WS – UNISIG KM WG held on 13-10-11	KM WG
0.1.1 15-02-12	§ 3.4.1.1; tables 4, 5, 7, 9 § 4.2.6.5, 4.2.6.5.1, 4.2.6.5.2	Updated following EEIG remarks. CR 814	
0.1.2 23-02-12	§ 3.2.1.1, 3.2.2, 3.3.1.1, 3.4.1.1 and front page	CR 758	
1.0.0 28-02-12	§1 and front page	Baseline 3 release version	KMS WG
1.0.1 15-12-15	§ 3.1.1.4, 3.3, 3.4, 4.6, 5.1.1.2 and Annex A	CR 1237	FH/PP
1.1.0 17-12-15	-	Baseline 3 2nd release version	PP
1.1.1 29-06-20	§6.2.1, 6.2.1.1, 6.3.1.9, 6.3.1.11,	Editorial changes according CR 1307	MK
3.9.2 23-02-23	§6.2.1, 6.2.1.1,	CR 1307 change reassigned to CR 1428	PL
4.0.0 05-07-2023	-	Baseline 4 1st release version	GP

2. TABLE OF CONTENTS

3. INTRODUCTION

3.1 Purpose and Applicability

3.1.1.1 ERTMS/ETCS applications are using open communication networks in order to transfer messages between trackside and on-board equipment (e.g. GSM-R PLMN) and also between trackside and trackside equipment (e.g. IP based RBC-RBC interface).

3.1.1.2 Data transmission links implemented over open networks are inherently vulnerable, as unauthorized access cannot be excluded. Therefore, it is important to ensure the integrity and authentication of messages being sent over the non-trusted transmission medium.

3.1.1.3 ETCS uses cryptographic techniques with secret keys to achieve this. Different ETCS specifications such as [Subset-037] and [Subset-098] assume that the required keys are already installed in the equipment. They do not describe how and in which format keys are transferred from the source (Key Management Centre) to the destination (ETCS entity) and installed.

3.1.1.4 The main purpose of Subset-114 is to harmonize the interface between the Key Management Centre (KMC) and the trackside and on-board ETCS equipment inside a key management domain in a policy-open way. This means the interface is not going to restrict railway operators from implementing a key management policy adequate for their special security needs (e.g. using different authentication keys for every ETCS pair, or using the same authentication key for a group of ETCS entities). In this subset, only an off-line interface is considered. This means that staff intervention is required to perform key management tasks, in particular to distribute key management requests to ETCS equipment.

3.1.1.5 This FIS is applicable to all ETCS entities (RBC, RIU, OBU) whose communication is based on secret keys and that therefore need to provide an interface for key installation, update and deletion purposes. The document is also applicable for KMC implementations that are going to perform key management tasks for ETCS equipment.

3.1.1.6 The following procedures and aspects are covered by this specification:

3.1.1.7 Chapter 4 describes the basic key management concepts and principles inside a key management domain.

3.1.1.8 Chapter 5 summarises required key management functions.

3.1.1.9 Key management messages are derived from these functions and the associated message structure is specified in chapter 6.

3.1.1.10 Annexes A and B contain an informative overview of operational KM scenarios for comprehensibility reasons.

3.1.1.11 This FIS does not mandate:

- Operational key management scenarios;
- Interoperability scenarios between different key management domains, including the KMC-KMC interface;
- Key management tasks to be performed by KMCs (e.g. key generation, storage, archiving etc.);
- The open networks used for ETCS entity communication and (on-line) key management tasks;
- The physical storage device to be used for distributing KM requests to an ETCS entity in case of off-line key management;
- The physical architecture of the cryptographic device inside an ETCS entity;
- The on-line interface between KMC and ETCS entity.

3.1.1.12 Fulfilling the requirements stated in this document contributes to achieving the security necessary for the protection of an ERTMS application against malicious intrusions. This document defines neither the security targets nor all additional measures that may be necessary to achieve them.

3.2 References

3.2.1 Normative References

3.2.1.1 This FIS incorporates by dated or undated references, provisions from other publications. 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.

[Subset-037]	ERTMS/ETCS; Subset-037; EURORADIO FIS
[Subset-038]	ERTMS/ETCS; Subset-038; Off-line Key Management FIS
[Subset-098]	ERTMS/ETCS; Subset-098; RBC-RBC Safe Communication Interface
[08E187]	08E187; ETCS Key Management System FRS, v.2, 23-10-2009
[EN 50159]	Safety related communication in open transmission systems. 09.10
[X3.92]	ANSI X3.92 1981 Data Encryption Standard (DES) algorithm
[X9.52]	ANSI X9.52 1998 Triple Data Encryption Algorithm Modes of Operation (parts only)
[ISO/IEC 9797-1]	Information technology - Security techniques – Messages Authentication Codes (MACs) - Part 1: Mechanisms using a block cipher, 12.99

3.2.2 Informative References

[04E518]	04E518; KMS Operational Aspects
[Subset-023]	ERTMS/ETCS; Subset-023; Glossary of Terms and Abbreviations
[Schneier]	Applied cryptography-Bruce Schneier-Wiley

3.3 Acronyms and Abbreviations

3.3.1.1 Please refer to the Glossary of Terms and Abbreviations [Subset-023] for more abbreviations.

CBC	Cipher Block Chaining
DES	Data Encryption Standard
ECB	Electronic Code Book Mode
KDC	Key Distribution Centre
KM	Key Management
KSMAC	Session Key
PLMN	Public Land Mobile Network
Triple-DES / 3DES	Triple-Data Encryption Standard

3.4 Terms and Definitions

3.4.1.1 Please refer to the Glossary of Terms and Abbreviations [Subset-023] for more terms and definitions.

Cryptography	The discipline which embodies the principles, means and methods for the transformation of data in order to hide its information content, prevent its undetected modification, prevent its unauthorised use or a combination thereof.
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 ([X3.92]).
DES-key	64 bits key respecting [X3.92].
ETCS Entity	ETCS OBU, RBC or RIU.
ETCS ID expanded	The unique identification of an ETCS entity of KMC, consisting of ETCS ID type and ETCS ID.
Key Entry	The triple-key itself with related information, i.e.: ⇨ identifier of the KMC that issued the key; ⇨ key serial number; ⇨ key validity period; ⇨ list of peer entities to which this key is allocated.
Key Deletion	Deletion of keys incl. all related information and copies.
Key Distribution	Confidential distribution of keys.
Key Distribution Centre	The optional functional entity, which is responsible to interface the Key Management Centre and the ETCS entities.
Key Generation	Confidential generation of keys used for encryption, decryption and key derivation.
Key Installation	Confidential installation of keys into ETCS entities.
Key Management	The generation, storage, secure distribution, revocation, deletion, archiving and application of keys/key entries in accordance with a security policy.
Key Store	The Key Store contains the key entries in the ETCS entity or the KMC.
Predefined key	Fixed triple-key used for authentication only, see §6.5. Predefined key.
Session Key	Triple-key, based on the authentication key, used for authentication of ERTMS entity during EuroRadio safe connection establishment.

Symmetric key	A cryptographic key that is used in symmetric keyed algorithms. A symmetric key is used for both encryption and decryption.
Transaction	A transaction is a request and related notification pair of messages.
Transport Key	Transport Key used to secure KMAC exchange between KMC and ETCS entities made of a couple of triple-keys KTRANS 1 and KTRANS 2 .
Triple-Key	Key composed of 3 concatenated DES keys, example : KTRANS 1, KTRANS 2, ….

4. KEY MANAGEMENT CONCEPTS AND PRINCIPLES

4.1 Introduction

4.1.1.1 In the ERTMS system, on board equipment and wayside RBCs or other related equipment exchange information using the EuroRadio protocol to secure communication over an open non-trusted medium.

4.1.1.2 When an ERTMS on-board equipment wishes to communicate with an RBC, it shall be able to verify that communication is established with an authorised RBC and vice versa. Consequently the authenticity and integrity of any information exchanged between ERTMS on-board equipment and RBC is also verified.

4.1.1.3 The method of ensuring that both communicating entities are the ones they state they are, is based on an Identification and Authentication (I&A) dialogue. In order to ensure complete protection, this procedure shall take place each time the peer entities effectively start a new safe connection.

4.1.1.4 After each successful I&A dialogue, data are protected using a Message Authentication Code (MAC). The calculation of this code is based on the existence of shared secret information known by the entities that are actually communicating with each other.

4.1.1.5 Both I&A dialogue and MAC calculation procedures are fully specified in the Safety Functional Module described in [Subset-037]. These procedures are based on particular cryptographic techniques that use secret triple-keys. However, they do not provide any means to create, distribute or update these triple-keys. Moreover, their full efficiency relies on the triple-key secrecy that can only be guaranteed when clear key management functions are defined according to implementation constraints and railway operational scenarios.

4.1.1.6 This specification only covers management and off-line distribution of cryptographic material between the Key Management Centre and the ETCS entities, based on symmetric algorithms.

4.2 Key hierarchy

4.2.1.1 The key hierarchy defined in [Subset-038] is applied in this symmetric key management specification.

Hierarchy level	Key name	Purpose
3	K-KMC	Transport key used for protection of KMS communication between KMCs
	KTRANS	Transport key used for protection of KMS communication between KMC and trackside or onboard equipment for installation, update or deletion of authentication keys.

2	KMAC	Authentication key used for session key derivation in order to establish a safe connection between two ETCS entities.
1	KSMAC	Session key used for protection of data transfer between two safety entities.

Table 1 – Key hierarchy

4.2.1.2 The following table summarises the usage of each type of key.

Involved entities	Key used for I & A	Key used for data protection	Key used for encryption	Interoperability
Trackside – on board	KMAC	KSMAC	Not applicable	relevant for interoperability
KMC – ETCS entity	Not applicable (off-line operations)	KTRANS1	KTRANS2	relevant for interoperability
KMC - KMC	Not applicable (off-line operations)	K-KMC1	K-KMC2	relevant for interoperability

Table 2 – Usage of keys

4.2.1.3 K-KMC and KSMAC keys are outside the scope of this specification.

4.2.2 Key Definitions

4.2.2.1 In this document a triple-key is defined as an array of 192 bits consisting of 3 concatenated DES-keys K1, K2, K3 of 64-bit length each.
In short: triple-key = K1 | K2 | K3, where the symbol "|" means concatenation.

4.2.2.2 Bit 0 is the left most bit of the triple-key.

4.2.2.3 For a valid triple-key, each eighth bit (LSB = bit 7, bit 15, …, bit 191) of the 192-bits must be set to an odd-parity value as defined in the standard [X3.92].

4.2.2.4 The following table shows the structure of a triple-key:

<table>
<tr><td colspan="3">triple-Key
length 192 bit</td></tr>
<tr><td colspan="3">b_0, b_1, … b_{191}</td></tr>
<tr><td>DES-key K1
length 64 bit</td><td>DES-key K2
length 64 bit</td><td>DES-key K3
length 64 bit</td></tr>
<tr><td>b_0, … b_{63}</td><td>b_{64}, … b_{127}</td><td>b_{128}, … b_{191}</td></tr>
</table>

4.2.3 Structure of KTRANS and KMAC

4.2.3.1 The KTRANS is defined as an array of 384 bits consisting of two concatenated triple-keys named KTRANS1 and KTRANS2.
In short: KTRANS = KTRANS1 | KTRANS2.

4.2.3.2 The KMAC is a triple-key.

4.2.4 Procedure for KMAC encryption and decryption

4.2.4.1 Each DES-key of the KMAC is encrypted and decrypted using the KTRANS2 according the Triple-DES process.

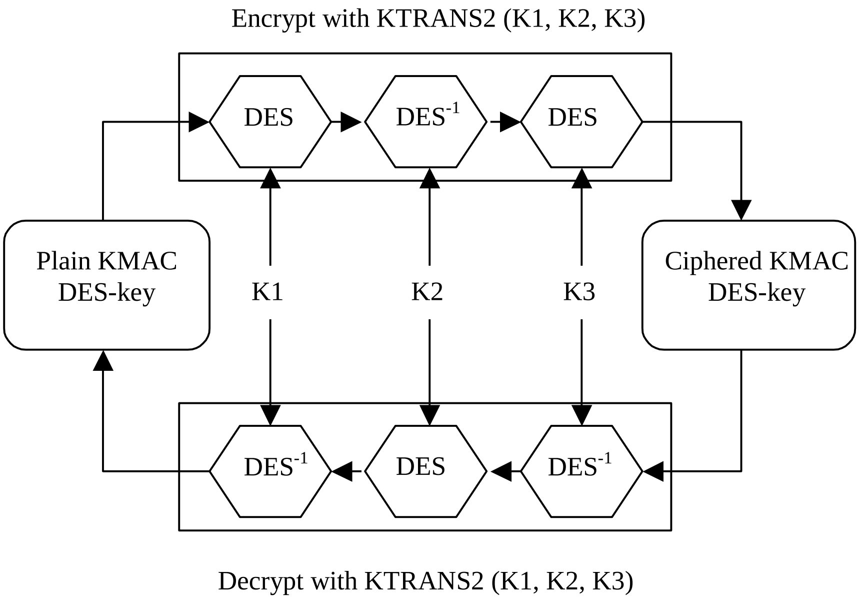


Figure 1: KMAC encryption using KTRANS2 (K1, K2, K3) and Triple-DES

4.2.4.2 The Triple-DES algorithm is used in mode ECB. Encryption/decryption corresponds to three DES operations in the sequence encrypt-decrypt-encrypt for encryption and decrypt-encrypt-decrypt for decryption, using keying option 1, i.e. three different DES-keys K1, K2, K3 of KTRANS2. (see [X9.52] § 6.1 & 6.2).

4.2.4.3 DES refers to the Data Encryption Algorithm as specified in [X3.92].

4.2.5 Procedure for CBC-MAC code calculation

4.2.5.1 The CBC-MAC is a value of 64 bits calculated on a message “m” using the KTRANS1 or a predefined key according to section “Generic MAC-Calculation” of [Subset-037]. Note: no other variables or values are added for the calculation of the CBC-MAC.

4.2.5.2 The purpose of the CBC-MAC is to check the integrity and to authenticate the message “m”.

4.2.6 Key validity period

4.2.6.1 The validity period shall be defined by the beginning date followed by the end date of validity period for the KMAC (in HH DD MM YY format e.g. 15 01 01 05 which means 1st January 2005 at 3:00 PM and BCD coded, 24 Hours format (from 00 to 23)).

4.2.6.2 The beginning date is included in the validity period, while the end date is excluded. Examples:

- beginning date “03 01 01 05” means that the key is valid from 3 AM, the 1st January 2005;
- end date “03 01 01 05” means that the key becomes invalid at 3 AM, the 1st January 2005.

4.2.6.3 UTC time shall be used for the interface.

4.2.6.4 The specific format 0xFFFFFFFF can be used for the end date only to specify infinite validity period.

4.2.6.5 Checking the key validity is specifiied in [Subset-037].

4.3 General principles

4.3.1.1 The following figure describes the architecture used for the Key Management System:

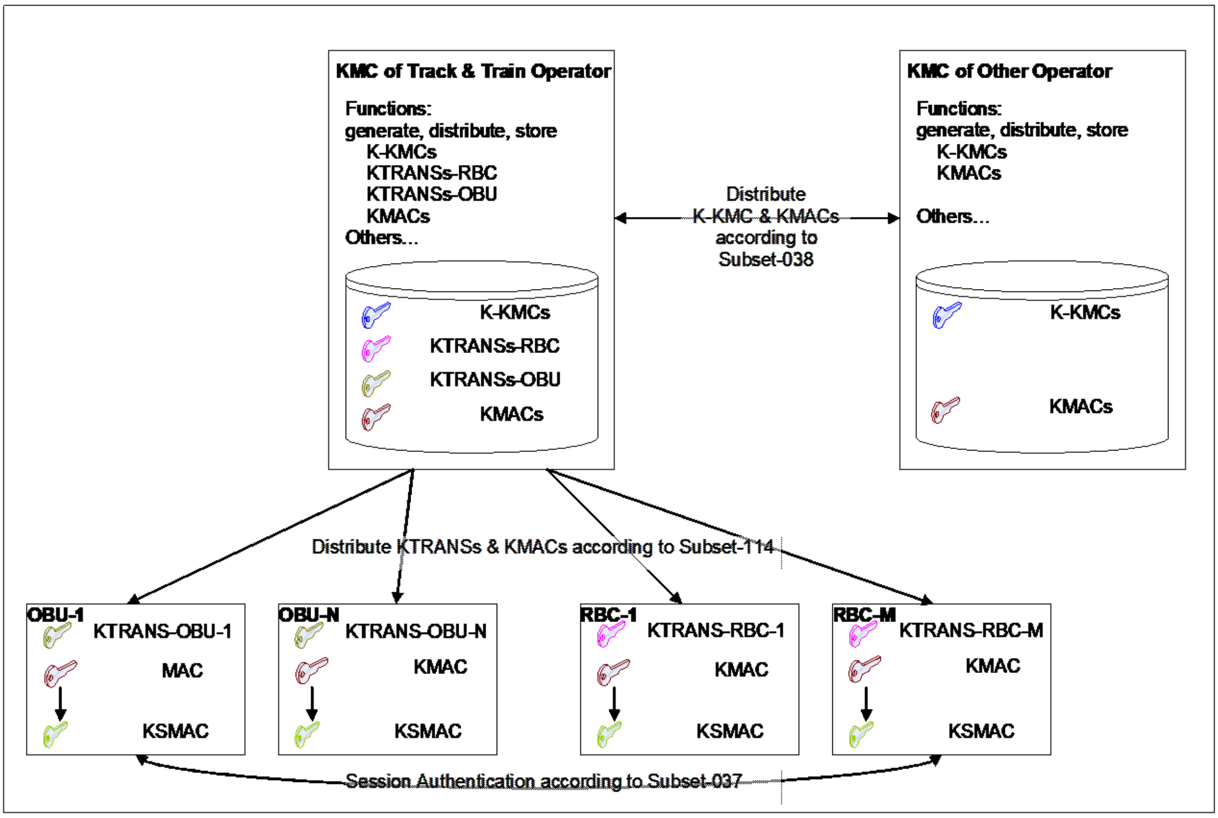


Figure 2: Key Management System architecture

4.3.1.2 Note : the symbols represent a type of key, not their quantity. For usage see the following description.

4.3.2 KMC-KMC interface

4.3.2.1 The KMC-KMC functional interface specifies the way to distribute the KMACs and K-KMCs between Key Management Domain, see [Subset-038].

4.3.3 Key management centre

4.3.3.1 One domain is defined as one KMC and all the on-board and trackside entities using that KMC (the Home KMC).

4.3.3.2 A KMC is responsible for the generation of the key entries needed to establish safe connections between trackside entities belonging to its domain and any on-board entity.

4.3.3.3 The KMC shall be able to uniquely identify all the generated keys.

4.3.3.4 Even if it is possible to allocate the same triple-key for KMACs attached to different onboard units, the key identifiers shall be different for each onboard unit.

4.3.3.5 A KMC is also responsible for distributing/updating/revoking KMAC to all trackside and onboard entities of its domain and interfacing to relevant KMCs.

4.3.3.6 The KMC shall ensure that the validity periods of two successive triple-keys applicable to the same relation are not overlapping.

4.3.3.7 KTRANS1 is used to ensure data protection between an entity and its Home KMC.

4.3.3.8 KTRANS2 is used to encipher KMAC keys sent from the Home KMC to its entities, for security purposes.

4.3.3.9 Each transaction between an ETCS entity and the Home KMC is initiated by the Home KMC.

4.3.3.10 Each request between a Home KMC and one of its ETCS entities is identified by a transaction number. This transaction number is only used to enable the KMC to match the notification with the related request.

4.3.3.11 The KMC is responsible for generating transaction numbers suitable for its purposes. The ETCS entity shall refer to the transaction number received with a KMC request in the associated notification message.

4.3.4 ETCS entity

4.3.4.1 The interface between ERTMS on-board and trackside equipment has been completely standardised for interoperability in [Subset-037].

4.3.4.2 Every on-board and trackside entity shall receive all needed keys from their Home KMC and each entity shall refer to only one Home KMC.

4.3.4.3 On-board and trackside entities shall use only their Home KMC for key management purposes.

4.3.4.4 Each ETCS entity expects to receive an unique identifier for each key distributed to it.

4.3.4.5 The unique identifier of the KMAC consists of the ETCS ID of the KMC that issued the key, and the key serial number.

4.3.4.6 The unique identifier of the KTRANS consists of the transport key serial number, the key being generated by the Home KMC.

4.3.4.7 Train supervision using ongoing ETCS connections shall not be affected by key management transactions.

4.3.4.8 An ETCS entity shall not modify any key entry installed by the Home KMC unless ordered to by the Home KMC.

4.3.4.9 The OBU shall be able to store 2000 key relations which consists of a relationship between a trackside unit, a KMAC and a validity period.

4.4 Functional Architecture

4.4.1.1 This subset specifies the interoperable interface to distribute keys between the KMC and the ETCS entities.

4.4.1.2 The following figure describes the functional architecture set up to implement the key distribution function to the ETCS entities:

K-KMC, KTRANS & KMAC in KMC, KDC & ETCS-Entity

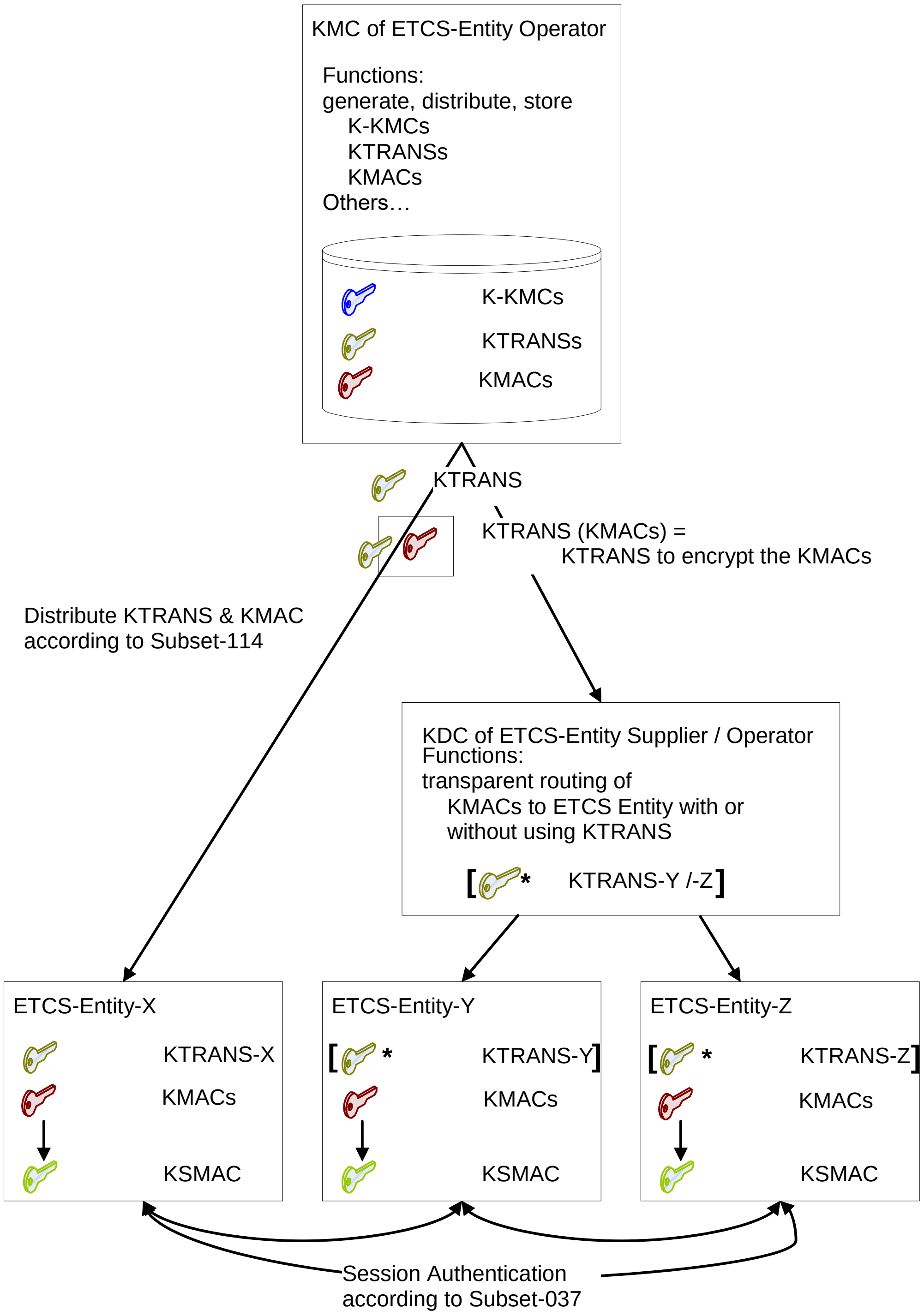


Figure 3: Functional architecture

4.4.1.3 Note : * KTRANS could be stored either in the KDC or in the ETCS entities.

4.4.1.4 Note : the symbols represent a type of key, not their quantity. For usage see the following description.

4.4.1.5 The key could be distributed directly from the KMC to the ETCS entity or, for message format migration or operational purposes, locally a Key Distribution Centre could be used to download keys into the ETCS entity.

4.4.1.6 The main function of the KDC is to distribute key data to the ETCS entities and to implement the standardised interface with the KMC. Therefore the KDC could fulfil the interoperable functions specified in this interface on behalf of the ETCS entity.

4.4.1.7 The optional use of a Key Distribution Centre shall be transparent for the interface with the KMC and shall not affect the KMC behaviour. This implies that:

- there is no ETCS ID allocated to the KDC;
- the messages sent by the KMC shall address directly the ETCS entity even if a KDC is present;
- the notifications sent back by the KDC shall address the KMC and shall report the ETCS entity identification as the issuer;
- the KDC must not modify any key entry installed by the Home KMC unless ordered to by the KMC.

4.5 Compliance to European standard

4.5.1.1 The key management and notification messages are sent using portable storage devices (e.g.: USB stick, hard disk, CD,…).

4.5.1.2 The key management and notification messages are sensitive data for which the authentication and confidentiality shall be ensured either by operational or technical measures.

4.5.1.3 For messages related to encrypted keys, the authentication and confidentiality of the key management and notification messages is ensured by technical measure (e.g. the use of the same secret key shared by the two end entities).

4.5.1.4 For messages related to unencrypted keys (e.g. KTRANS), the authentication and confidentiality of the key management and notification messages has to be ensured by operational measures.

4.5.1.5 The EuroRadio protocol complies with [EN 50159]. To maintain the compliance, railways key management must enforce key confidentiality. This standard contributes to this by ensuring authenticity and confidentiality (MAC code and key encryption) in the key distribution process.

4.5.1.6 Because the key management is security related rather than safety related, the only threat of the standard [EN 50159] that applies to this interface is masquerade. This is addressed through MAC code and key encryption.

4.5.1.7 Nevertheless the other threats of the standard [EN 50159] are detected in the following way:

- Message repetition: KMC or the ETCS entity receives several messages with the same transaction number;
- Message deletion: KMC does not receive the expected notification message;
- Message insertion: Source and Destination Identifier are included in the message header (a message coming from a KMC other than the Home KMC shall be rejected by the ETCS entity);
- Message resequencing: Message sequence number is included in the message header;
- Data corruption: MAC code is included in the message structure;
- Delay: KMC does not receive or receives too late the notification message;

4.5.1.8 In the case of this off-line key management process, as the specification of the key distribution process is not part of this standard, some of the threats related to key distribution could also be covered by a national procedure.

4.6 Intentionally deleted

5. KEY MANAGEMENT FUNCTIONS

5.1 Introduction

5.1.1.1 The following section specifies the basic key management functions that are required for efficient key management between KMC and ETCS entities. Additional functions could exist locally but shall not interfere with this specification.

5.1.1.2 Intentionally deleted.

5.1.1.3 Two different key handling methods are defined for the installation, update and removal of keys:

a) with the “all” key handling method a full set of keys is installed in an ETCS entity if the KMC wants to add, modify or delete keys. The key store of this entity is automatically cleared before a change;

b) with the “single” key handling method a KMC requests the installation, modification or deletion of a single key in an ETCS entity.

5.1.1.4 Each ETCS entity shall implement at least one of these methods. The KMC has to know the handling method used by each ETCS entity in its domain and shall issue the corresponding compatible request..

5.1.1.5 The following table specifies for each key handling methods, the functions to be supported by the ETCS entity. “M” and “NA” indicate respectively if the function is “Mandatory supported” or “Not Applicable”.

Functions	**“ single” key handling method**	**“all” key handling method**
Install Transport Key function	“M”	“M”
Replace All Authentication Keys	“NA”	“M”
Delete All Keys	“M”	“M”
Add Authentication Key	“M”	”NA”
Delete Authentication Key	“M”	”NA”
Replace ETCS Entities	“M” for Onboard units “NA” for trackside units	“NA”
Update key validity period	“M”	”NA”

Table 3 – Key handling methods

5.2 Install Transport Key

5.2.1.1 This function is used to distribute a transport key from the KMC to ETCS entities.

5.2.1.2 The current subset specifies only the message format to be used to distribute unencrypted KTRANS. As the transport key is not enciphered, the message protection during the distribution shall be ensured by operational procedures that are out of scope of this document.

5.2.1.3 This function is an off-line function.

5.2.1.4 When a new KTRANS is installed in an ETCS entity, the old one is replaced, if present.

5.2.1.5 The installation by itself shall not impact the validity of already-installed authentication keys in the ETCS entity.

5.3 Replace All Authentication Keys

5.3.1.1 This function shall be used by the KMC to replace, for an ETCS entity, the full set of authentication keys under the responsibility of the KMC.

5.3.1.2 The effect of this function is to delete all the authentication keys already installed in the ETCS equipment before performing the installation of the new key set.

5.3.1.3 The appropriate transport key is needed to perform the installation.

5.4 Delete All Keys

5.4.1.1 This function shall be used by the Home KMC to delete, in the ETCS entity, the different types of keys installed by the Home KMC.

5.4.1.2 The deletion of the following key types shall be supported :

- all authentication keys installed by the Home KMC;
- the transport key installed by the Home KMC;
- all keys distributed by the KMC to the ETCS entity, including the transport key;

5.4.1.3 The deletion process has to be performed in such way that the deleted keys could not be recovered.

5.4.1.4 The appropriate transport key is needed to perform the deletion.

5.4.1.5 If the deletion of the transport key is requested, the notification message shall be authenticated using the predefined key.

5.5 Add Authentication Key

5.5.1.1 This function shall be used by the KMC to add a single authentication key to an ETCS entity.

5.5.1.2 The appropriate transport key is needed to perform the installation.

5.6 Delete Authentication Key

5.6.1.1 This function shall be used by the KMC to remove a single authentication key from an ETCS entity.

5.6.1.2 The deletion process has to be performed by the ETCS entity in such a way that the deleted key could not be recovered.

5.6.1.3 The ETCS entity shall never delete an authentication key on its own, even in case of an outdated key. The deletion request shall always be ordered by the Home KMC.

5.6.1.4 The appropriate transport key is needed to perform the deletion.

5.7 Replace ETCS Entities

5.7.1.1 This function shall be used by the KMC to replace the list of peer ETCS entities of an installed authentication key. At least one ETCS peer entity shall be listed in the request.

5.7.1.2 At the reception of the request, the ETCS entity shall replace the list of ETCS entities linked to an authentication key with the list included in the request.

5.7.1.3 This function is applicable only to onboard entities.

5.7.1.4 The appropriate transport key is needed to perform the replacement.

5.8 Update key validity period

5.8.1.1 This function shall be used to update the key validity period (start and end dates) of a triple-key already installed in the ETCS entity.

5.8.1.2 The update of the validity period of a triple-key shall not affect the other key properties.

5.8.1.3 The appropriate transport key is needed to perform the update.

5.8.1.4 It is the KMC's responsibility to ensure that no keys with overlapping periods are installed in the ETCS entity.

6. KEY MANAGEMENT TRANSACTIONS

6.1 Introduction

6.1.1.1 This section defines the structure of the transaction messages exchanged between KMC and ETCS entities/KDC. Two messages types have to be distinguished:

- key management requests;
- key management notification messages.

6.1.1.2 Key management requests are generated in the KMC and submitted to an ETCS entity/KDC. The ETCS entity/KDC responds to requests by means of notification messages. These notification messages either acknowledge the request reception or they contain the positive or negative processing result.

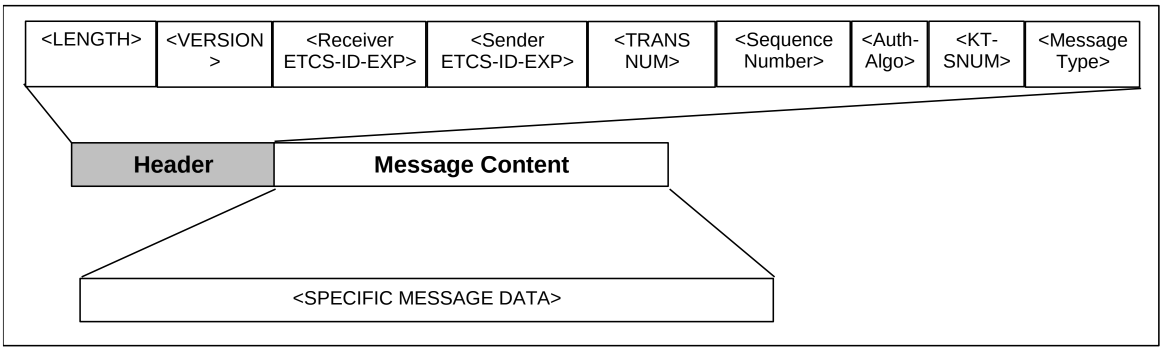
6.1.1.3 A unique identifier consisting of ETCS ID and ETCS ID type is assigned to each ETCS entity and KMC (see [Subset-037]"Transport Layer Addressing"). The phrase "ETCS ID expanded" is used as a reference to this unique identifier in the following chapters. The coding is as follows:

8765 4321 (bit)	Encoding Fields
xxxx xxxx	ETCS ID type (1 octet)
yyyy yyyy yyyy yyyy yyyy yyyy	ETCS ID (3 octets)

Table 4 – Encoding of ETCS ID expanded

6.1.1.4 All messages are specified in binary format, using big-endian representation.

6.1.1.5 The general message structure is depicted in the figure below.



6.1.1.6 The global message header consists of length, version, unique identification of receiver and sender (ETCS ID expanded), transaction number, sequence number, authentication algorithm, transport key serial number and message type.

© This document has been developed and released by UNISIG

6.1.1.7 The next table illustrates the structure of the message header.

Notes:

a) In the following tables, the term “undefined” means that the values can be used for local implementation but there may be interworking issues. The term “reserved” means that the values are reserved for use within the scope of this document.

b) The interface provides the possibility to detect and report out of sequence messages but it does not prevent the message processing in case of sequence errors. The decision whether an out of sequence message is an error or not depends on the particular circumstances (e.g. key handling method, request type, independence of single commands) and can only be taken at the KMC level. If the interface strictly prevents the processing of out of sequence messages this could lead to unnecessary operational obstructions and delays.

Table 5 Structure of the Message Header

Octet	Bit 8765 4321	Field	Description
1 2 3 4	xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx	LENGTH	Total length of the message including the length field.
5	xxxx xxxx 0000 0001	Version Current version	Version of the interface
6 7 8 9	xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx	Receiver ETCS-ID-EXP	The unique identification of the receiver contains the ETCS ID expanded of the ETCS entity in the case of key management requests, and of the KMC for notifications
10 11 12 13	xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx	Sender ETCS-ID-EXP	The unique identification of the message sender contains the ETCS ID expanded of the KMC in the case of key management requests, and the ETCS ID of the processing entity for notifications
14 . . 17	xxxx xxxx . . xxxx xxxx	TRANS NUM	The transaction number enables the KMC to establish request - notification relations
18 19	xxxx xxxx xxxx xxxx 0000 0000 0000 0000	Sequence number Sequence number ignored	The sequence number makes it possible to detect when requests are handled in the wrong sequence. An error in sequence number does not prevent the ETCS entity from executing the request The ETCS entity expects a sequence number which is equal to the sequence number included in the last received request incremented by one except in the following cases: - the sequence number wraps around to 1 after 0xFFFF - the sequence number value 0x0000 shall be used only to indicate that the sequence number shall be

			ignored
20	xxxx xxxx 0000 0000 0000 0001 0xxx xxxx 1xxx xxxx	Auth-ALGO No authentication algorithm used (not used in current release) Single DES with modified MAC algorithm 3 as defined ISO9797-1 Reserved Undefined	Algorithm used for message authentication In this release only the use 'single DES with modified MAC algorithm 3 as defined ISO9797-1' is requested. All the CBC-MAC fields defined in the following messages are computed using this algorithm
21 . . 24	xxxx xxxx . . xxxx xxxx 0000 0000 0000 0000 0000 0000 0000 0000	KT-SNUM Used for predefined key for MAC computation	Serial number of KTRANS that has been used for MAC calculation and encryption Note: use of predefined key ensures only integrity
25	xxxx xxxx 0000 0001 0000 0010 0000 0011 0000 0100 0000 0101 0000 1000 0000 1001 0100 0001 0xxx xxxx 1xxx xxxx	Message Type REPLACE_ALL_KEYS DELETE_ALL_KEYS ADD_AUTHENTICATION_KEY DELETE_KEY REPLACE_ETCS_ENTITIES UPDATE_KEY_VALIDITY_PERIOD INSTALL_TRANSPORT_KEY RESPONSE_NOTIF Reserved Undefined	This parameter identifies the key message type

6.2 Format of Key Management Requests

6.2.1 REPLACE_ALL_KEYS

Table 6 Structure of REPLACE_ALL_KEYS Request

Octet	Bit 8765 4321	Field	Description
26	xxxx xxxx 0000 0001 0xxx xxxx 1xxx xxxx	E-ALGO CRYPT-3DES-ECB reserved Undefined	Algorithm used for KMAC encryption/decryption
27 28	xxxx xxxx xxxx xxxx	K-NUM	Number z of triple-keys contained in the structure. K-NUM shall be greater than or equal to '1'

29 . . . n	xxxx xxxx . . . xxxx xxxx	K-STRUCTk	Key structure containing the triple-key itself and all additional properties/parameters (k=1 to z).
n+1 . . . n+8	xxxx xxxx . . . xxxx xxxx	CBC-MAC	The CBC-MAC shall be calculated over the complete message from octet 1 up to but excluding the CBC-MAC field using transport key KTRANS1

6.2.1.1 The key structure K-STRUCT consists of the following parameter

Table 7 Structure of K-STRUCT

Octet	Bit 8765 4321	Field	Description
p	xxxx xxxx	K-LENGTH (kl)	Key length in octets (24 for KMAC) Note: E-ALGO identifies only the encryption algorithm but not the type of key encrypted
p+1 . . p+4	xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx	KM-ETCS-ID-EXP	Unique identification of the KMC that issued the authentication key
p+5 . . p+8	xxxx xxxx . . xxxx xxxx	SNUM	Unique serial number of the key (together with the KM-ETCS-ID-EXP, this identifies the triple-key unambiguously) The upper 8 bits shall not be used and be set to '0' in compliance with Subset-038.
P+9 . . . p+8+kl	xxxx xxxx . . . xxxx xxxx	ENC(KMAC)	Authentication key encrypted with transport key KTRANS-2 (see § 4.2)
p+9+kl p+10+kl	xxxx xxxx xxxx xxxx	PEER-NUM	Number j of peer (on-board or trackside) ETCS entities stored in the structure. PEER-NUM shall be greater or equal to ‘1’ Note : Trackside and on-board ETCS entities should not be mixed in the same structure
p+11+kl+4*(i-1) . . p+14+kl+4*(i-1)	xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx	PEER-ETCS-ID-EXPi	ETCS ID Expanded of peer entity stored in the structure (i = 1 to j) Note : Trackside and on-board ETCS entities should not be mixed in the same structure
p+15+kl+4*(j-1) . . . . . p+22+kl+4*(j-1)	xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx	VALID-PERIOD	Validity period (see § 4.2 for format definition)

6.2.2 DELETE_ALL_KEYS

Table 8 Structure of DELETE_ALL_KEYS Request

Octet	Bit 8765 4321	Field	Description
26	xxxx xxxx	Key type	This field indicates the type of key to be deleted. Only the keys distributed by the KMC shall be considered
	0000 0001	KMAC	
	0000 0010	KTRANS	
	0000 0011	KMAC + KTRANS	
	0xxx xxxx	Reserved	
	1xxx xxxx	Undefined	
27 . . . 34	xxxx xxxx . . . xxxx xxxx	CBC-MAC	The CBC-MAC shall be calculated over the complete message from octet 1 up to but excluding the CBC-MAC field using transport key KTRANS1

6.2.3 ADD_AUTHENTICATION_KEY

Table 9 Structure of ADD_AUTHENTICATION_KEY Request

Octet	Bit 8765 4321	Field	Description
26 . . . 48+kl+4*(j-1)	xxxx xxxx . . . xxxx xxxx	K-STRUCT	Key structure as described in § 6.2.1.1.
49+kl+4*(j-1) . . . 56+kl+4*(j-1)	xxxx xxxx . . . xxxx xxxx	CBC-MAC	The CBC-MAC shall be calculated over the complete message from octet 1 up to but excluding the CBC-MAC field using transport key KTRANS1

6.2.4 DELETE_KEY

Table 10 Structure of the DELETE_KEY Request

Octet	Bit 8765 4321	Field	Description
26 27 28 29	xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx	KM-ETCS-ID-EXP	ETCS ID type and ETCS ID of the KMC that issued the authentication key to be deleted
30 . . 33	xxxx xxxx . . xxxx xxxx	SNUM	Unique serial number of key to be deleted (together with the KM-ETCS-ID-EXP, this identifies the triple-key unambiguously)

34 . . . 41	xxxx xxxx . . . xxxx xxxx	CBC-MAC	The CBC-MAC shall be calculated over the complete message from octet 1 up to but excluding the CBC-MAC field using transport key KTRANS1

6.2.5 REPLACE_ETCS_ENTITIES

Table 11 Structure of the REPLACE_ETCS_ENTITIES Request

Octet	Bit 8765 4321	Field	Description
26 27 28 29	xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx	KM-ETCS-ID-EXP	ETCS ID type and ETCS ID of the KMC that issued the authentication key to which entities are replaced
30 . . 33	xxxx xxxx . . xxxx xxxx	SNUM	Unique serial number of the authentication key to which entities are to be added (together with the KM-ETCS-ID-TYPE, this identifies the triple-key unambiguously)
34 35	xxxx xxxx xxxx xxxx	PEER-NUM	Number m of peer ETCS entities. PEER-NUM shall be greater or equal to '1'
36+(i-1)*4 . . 39+(i-1)*4	xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx	PEER-ETCS-ID- EXPi	ETCS ID TYPE and ETCS ID of peer ETCS entities (i=1 to m)
40+(m-1)*4 . . . 47+(m-1)*4	xxxx xxxx . . . xxxx xxxx	CBC-MAC	The CBC-MAC shall be calculated over the complete message from octet 1 up to but excluding the CBC-MAC field using transport key KTRANS1

6.2.6 UPDATE_KEY_VALIDITY_PERIOD

Table 12 Structure of the Update validity period Request

Octet	Bit 8765 4321	Field	Description
26 27 28 29	xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx	KM-ETCS-ID-EXP	ETCS ID type and ETCS ID of the KMC that issued the authentication key to be updated
30 . . 33	xxxx xxxx . . xxxx xxxx	SNUM	Unique serial number of the triple-key of which the key validity period has to be updated (together with the KM-ETCS-ID-EXP, this identifies the triple-key unambiguously)
34 41	xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx	VALID-PERIOD	Updated validity period(see § 4.2 for format definition)
42	xxxx xxxx	CBC-MAC	The CBC-MAC shall be calculated over the

. . . 49	. . . xxxx xxxx		complete message from octet 1 up to but excluding the CBC-MAC field using transport key KTRANS1

6.2.7 INSTALL_TRANSPORT_KEY

Table 13 Structure of the Install Transport Key Request

Octet	Bit 8765 4321	Field	Description
26	xxxx xxxx	KT-LENGTH (kl1)	Total length of the transport keys (48 octets for KTRANS1 + KTRANS2)
27 . . 30	xxxx xxxx . . xxxx xxxx	KT-SNUM	Serial number of the distributed KTRANS. The transport key serial number ‘0’ is reserved for the predefined key and shall not be used to install a transport key
31 . . . p p+1 . . . 30+kl1	xxxx xxxx . . . xxxx xxxx xxxx xxxx . . . xxxx xxxx	KTRANS1 KTRANS2	Unencrypted KTRANS1 Unencrypted KTRANS2
31+kl1 . . . 38+kl1	xxxx xxxx . . . xxxx xxxx	CBC-MAC	The CBC-MAC shall be calculated over the complete message from octet 1 up to but excluding the CBC-MAC field using the predefined key

6.2.7.1 For the header of the “Install Transport Key” request, the values of the “KT-SNUM” field shall be set to “Reserved for predefined key used for MAC computation".

6.2.7.2 The predefined key is not secret and is used only for message integrity check. The protection of the “Install Transport Key Request” shall be ensured by operational procedure.

6.3 Notification Messages

6.3.1.1 After reception and processing of a key management request an ETCS entity returns a notification message to the KMC.

6.3.1.2 Every request is answered by a response notification.

6.3.1.3 The successful processing of the request is reported using the return code “success”.

6.3.1.4 If the request processing is delayed, the reception of the request can be reported using the return code “reception successful” (e.g. delay due to the use of KDC). This is only an indication reported to the KMC and the entity still has to return the result of the request processing.

6.3.1.5 Processing failures, reception of unknown or unsupported requests are indicated by the ETCS entity using a response notification with a return code other than “success” or “reception successful”.

6.3.1.6 Upon receipt of an unknown or unsupported request, a failure shall be indicated by the ETCS entity.

6.3.1.7 The transaction number of the notification shall be set to the transaction number of the corresponding request in order to establish request - notification relations.

6.3.1.8 The sequence number included in the header of the notification message shall be set to the sequence number of the corresponding request.

6.3.1.9 The response notification consists of the following parameters:

Table 14 Structure of RESPONSE_NOTIF Message

Octet	Bit 8765 4321	Field	Description
26	xxxx xxxx	RESULT	Result of request execution or confirmation of the reception. For retuned code definition see next table Confirmation of reception is optional
27	xxxx xxxx	Length of optional additional text field (LT)	Set to 0 if text field is not used, maximum text length : 255 octets
28 ... 27+ LT	Additional text	ADDTEXT	Text in ASCII format(each character coded in 1 octet, most significant bit being the bit `0´) , if any
27 + LT + 1 27 + LT + 2	xxxx xxxx xxxx xxxx	Expected sequence number	The ETCS entity expects a sequence number which is equal to the sequence number included in the last received request incremented by one except in the following cases: - the sequence number wraps around to 1 after 0xFFFF;

			- sequence number value 0x0000 shall be used only to indicate that the sequence number shall be ignored The comparison between the ´Expected sequence number` and the `Sequence number` included in the header allows the detection of message sequence errors
27 + LT + 3 . . . 27 + LT +10	xxxx xxxx . . . xxxx xxxx	CBC-MAC	The CBC-MAC shall be calculated over the complete message from octet 1 up to but excluding the CBC-MAC field using transport key KTRANS1 or the predefined key KTRANS1 shall be used for the MAC computation when it is possible Only when it is not possible the predefined key has to be used (e.g. : KTRANS missing) Note : the KTRANS serial number reported in header of the notification message shall be consistent with the key used to compute the MAC field

6.3.1.10 The following table defines, for each result reported in the response notification message, the applicability, code value and presence or not of the MAC code.

Code value (dec)	Description	DELETE_ALL_KEYS	REPLACE_ALL_KEYS	ADD_AUTHENTICATION_KEY	UPDATE_KEY_VALIDITY PERIOD	DELETE_KEY	REPLACE_ETCS_ENTITIES	INSTALL_TRANSPORT_KEY
0	Request succesfully processed	x	x	x	x	x	x	x
1	Request received succesfully	x	x	x	x	x	x	x
2	Authentication of MAC code has failed	x	x	x	x	x	x	x
3	Authentication algorithm not implemented	x	x	x	x	x	x	x
4	Transport key not found	x	x	x	x	x	x	
5	Decryption algorithm not implemented		x	x				
6	Key not known				x	x	x	
8	Maximum number of keys exceeded		x	x				
9	Maximum number of ETCS entities exceeded		x	x			x	
10	Key already defined in the ETCS entity			x				
11	Request not supported		x	x	x	x	x	
12	Inconsistency detected in the received request. Note that this error code has to be used for all inconsistencies that may be detected in the message, except for the “message length" error.	x	x	x	x	x	x	x
13	Message length error	x	x	x	x	x	x	x
14	Request not issued by the Home KMC	x	x	x	x	x	x	x
15	Request sent to wrong ETCS entity	x	x	x	x	x	x	x
16	Key corrupted (parity bit or other key related consistency check failed)		x	x				x

17	Unrecoverable key store error	x	x	x	x	x	x	x
18	Interface version not supported	x	x	x	x	x	x	x
127	To be used for all errors not defined above	x	x	x	x	x	x	x

6.3.1.11 The error codes 7 and from 19 to 126 are reserved.

6.3.1.12 The error codes from 128 to 255 are undefined.

6.4 Structure of off-line transport medium

6.4.1.1 This specification does not define the transport medium between the KMC and the ETCS entity or KDC. However it is possible to define a top level entry point (e.g. root directory in an USB stick, specific directory on hard disk, …).

6.4.1.2 This top level entry point shall contain a directory for each addressed ETCS entity, formatted as follow: “tteeeeee” (with t, e out of {0..9, A..F}). Where:

- ⇨ *tt* indicates the ETCS ID type of the ETCS entity to which the request is dedicated in hexadecimal format;
- ⇨ *eeeeee* indicates the ETCS ID of the ETCS entity to which the request is dedicated in hexadecimal format.

6.4.1.2.1 Example : an RBC with ETCS ID decimal 169 would be hexadecimal “010000A9”.

6.4.1.3 All the requests related to an ETCS entity are directly provided in the directory dedicated to the ETCS entity.

6.4.1.4 As a consequence of the directory structure, the requests related to different ETCS entities shall be provided in different directories.

6.4.1.5 Each request shall be provided in a different file in binary format.

6.4.1.6 The file name shall be formatted as follow: “yymmddhhmmsszzzzzz.req”. Where:

- ⇨ *yy* indicates the year when the request has been generated;
- ⇨ *mm* indicates the month when the request has been generated;
- ⇨ *dd* indicates the day of the month when the request has been generated;
- ⇨ *hhmmss* indicate respectively the hour of the day, the minute and the second when the request has been generated, using 24-hour format;
- ⇨ *zzzzzz* is a number allowing the specification of the order of different requests to the same ETCS entity and generated within the same second: fixed decimal format, left padded with “0”;
- ⇨ .req shall be the extension used for the request files.

6.4.1.7 The filename indicates the order of generation of the request in the KMC. The requests shall be processed in the same order as they have been generated.

6.4.1.8 To each request file corresponds a notification file to report the result of the request processing and optionaly a notification to report the request reception. Each notification shall be stored in a different file.

6.4.1.9 All the notifications related to an ETCS entity are provided in a directory dedicated to that ETCS entity.

6.4.1.10 As a consequence of the directory structure, notifications related to different ETCS entities shall be provided in different directories.

6.4.1.11 The notification message that reports the result of the request processing shall reuse the request file name to which it corresponds with extension *.rsp* instead of *.req*.

6.4.1.12 The notification message that reports the reception of the request shall reuse the request file name to which it corresponds with extension *.rxd* instead of *.req*.

6.4.1.13 All the names for requests and notifications shall be formatted using lower case.

Note: depending on the technology used, the medium used to report the notifications could be different from the one used to transmit the requests.

6.5 Predefined key

6.5.1.1 This section details the predefined triple-key. It has a length of 192 bits, and consists of three 64 bit DES keys.

6.5.1.2 The triple-key must be distributed as defined by [X9.52], where each eighth bit (the LSB) is defined as an odd-parity bit. Implementations may or may not check this parity bit.

6.5.1.3 The first valid DES key (bits 0 to 63) is:

```
MSB            LSB   hex
0 0 0 0  0 0 0  1    0 1
0 0 0 0  0 0 1  0    0 2
0 0 0 0  0 1 0  0    0 4
0 0 0 0  0 1 1  1    0 7
0 0 0 0  1 0 0  0    0 8
0 0 0 0  1 0 1  1    0 B
0 0 0 0  1 1 0  1    0 D
0 0 0 0  1 1 1  0    0 E
```

6.5.1.4 The format for key distribution, using this DES key as an example, will be as follows, with the greatest-weight bit being b0, b8, b16 ..., and each parity bit being b7, b15, b23 Note that '|' is used here as the concatenation operator.

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

or 01 | 02 | 04 | 07 | 08 | 0B | 0D | 0E

6.5.1.5 The second DES key (bits 64 to 127) 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
```

6.5.1.6 The third DES key (bits 128 to 191) 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
```

6.5.1.7 The final predefined triple-key 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
```

6.5.1.8 To test a correct implementation, the following check may be used, using the triple-DES process with the predefined key in ECB mode:

plaintext	01 23 45 67 89 AB CD EF
K1	01 02 04 07 08 0B 0D 0E
encrypt >	35 47 E5 88 7D D6 AB A4
K2	10 13 15 16 19 1A 1C 1F
decrypt >	AA 8E 03 4F B6 4D 14 6F
K3	20 23 25 26 29 2A 2C 2F
encrypt >	C3 CF BC A7 E2 49 17 82

This final result is the ciphertext.

Annex A (INFORMATIVE) OPERATIONAL KEY MANAGEMENT SCENARIOS

A.1.1.1 This chapter describes the scenarios that have to be performed by a key management centre (KMC) within a key management domain in order to install, update and delete keys in an ETCS entity (OBU, RBC, RIU).

A.1.1.2 The introduction and decommissioning of ETCS entity (e.g. OBU) in a foreign key management domain is out of scope of this document, due to the fact that the installation, update and removal of keys is always performed by an ETCS entity's Home KMC according to the scenarios described in the subsequent chapters.

A.1.1.3 Annex B contains some scenarios regarding the cross-domain key handling for comprehensibility reasons.

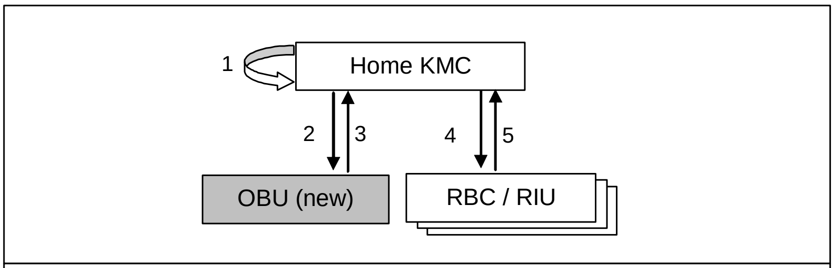
A.1.1.4 Two different key handling methods are defined for the installation, update and removal of keys:

- With the "all" key handling method a full set of keys is installed in an ETCS entity if the KMC wants to add, modify or delete keys. The key store of this entity is automatically cleared before any change;
- With the "single" key handling method a KMC requests the installation, modification or deletion of a single key in an ETCS entity.

A.1.1.5 Both methods and the associated request sets are described in more detail in § 5.1.

A.2 Introduction of new on-board entity into service

A.2.1.1 The next picture shows the actions that have to be performed by the KMC when introducing a new OBU in its home key management domain. In this example, the same authentication key is used to protect the communication of this OBU with each RBC / RIU of the domain.

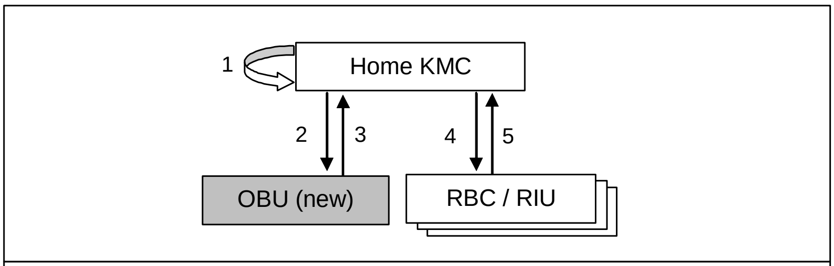


1) Generate a new key and store it in the key store
2) “Single” key / “all” key handling method in new OBU: “Add Authentication Key”/ “Replace All Authentication Keys”
3) Notify KMC about result

For 1 … number of trackside units (RBCs / RIUs):
4) “Single” key handling method in RBC/RIU: “Add Authentication Key”
“All” key handling method in RBC/RIU: “Replace All Authentication Keys”
5) Notify KMC about result

Note 1: the steps 2 and 4 can be done either in parallel or consecutively.
Note 2: all OBUs that are already in service remain unchanged.

A.2.1.2 The next picture illustrates the actions that have to be performed by the KMC when introducing a new OBU in its home key management domain. In this example, the KMC uses a key management policy that assigns a different authentication key to each OBU – RBC and OBU – RIU relation.



1) For k = 1 … number of trackside units (RBCs / RIUs):
Generate a new key and store it in the key store

“Single” key handling method in new OBU:
for k = 1 … number of trackside units:
2) “Add Authentication Key”
3) Notify the KMC about result

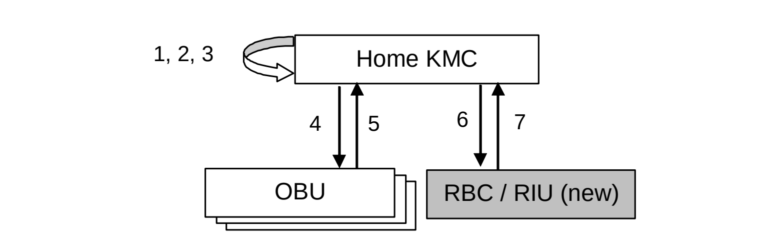
“All” key handling method in new OBU:
2) “Replace All Authentication Keys”
3) Notify KMC about result

For k = 1 … number of trackside units:
4) “Single” key handling method in RBC/RIU k: “Add Authentication Key”
“All” key handling method in RBC/RIU k: “Replace All Authentication Keys”
5) Notify KMC about result

Note 1: the steps 2 and 4 can be done either in parallel or consecutively.
Note 2: all OBUs that are already in service remain unchanged.

A.3 Introduction of new trackside entity into service

A.3.1.1 This picture shows the actions that have to be performed by the KMC when introducing new trackside entity, e.g. RIU or RBC, in the home key management domain. In this example, the same authentication key is used to protect the communication of a particular OBU with each RBC / RIU of this domain.



For k = 1 … number of OBUs:

1) Retrieve key related to OBU k from key store
2) Add identifier of new RBC / RIU to trackside list of key
3) Store updated key in the key store
4) "Single" key handling method in OBU: "Replace ETCS Entities" list of stored key
   "All" key handling method in OBU: "Replace All Authentication Keys"
5) Notify KMC about result

"Single" key handling method in new RBC/RIU:

For k = 1 … number of OBUs:

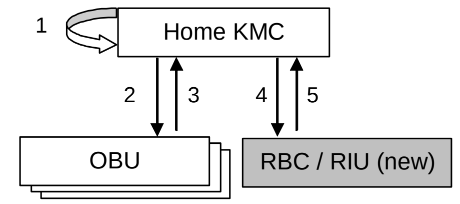
6) "Add Authentication Key"
7) Notify KMC about result

"All" key handling method in new RBC/RIU:

6) "Replace All Authentication Keys"
7) Notify KMC about result

Note: all RBCs / RIUs that are already in service remain unchanged.

A.3.1.2 The next picture illustrates the actions that have to be performed by the KMC when introducing new trackside entity, e.g. RIU or RBC, in the home key management domain. In this example, the KMC uses a key management policy that assigns a different authentication key to each OBU – RBC and OBU – RIU relation.



For k = 1 … number of OBUs:

1) Generate new key and store it in key store
2) "Single" key handling method in OBU k: "Add Authentication Key"
   "All" key handling method in OBU k: "Replace All Authentication Keys"
3) Notify KMC about result

"Single" key handling method in new RBC/RIU:

for k = 1 … number of OBUs:
4) "Add Authentication Key"
5) Notify KMC about result

"All" key handling method in new RBC / RIU:

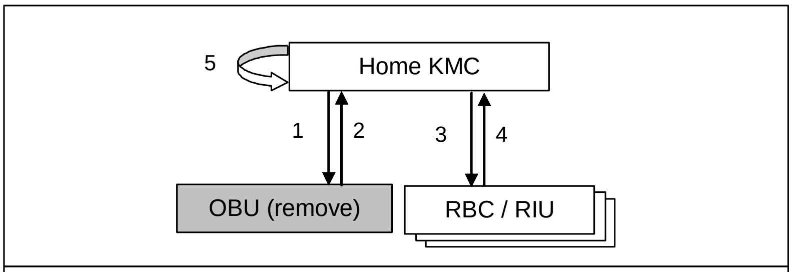
4) "Replace All Authentication Keys"
5) Notify KMC about result

Note 1: the steps 2 and 4 can be done either in parallel or consecutively.

Note 2: all RBCs / RIUs that are already in service remain unchanged.

A.4 Decommissioning of on-board entity

A.4.1.1 This picture shows the actions that have to be performed by the KMC when an OBU goes out of commission in its home key management domain. In this example, the same authentication key has been used to protect the communication of this OBU with each RBC / RIU located in the domain.



1) "Delete All Keys" from the OBU
2) Notify KMC about result

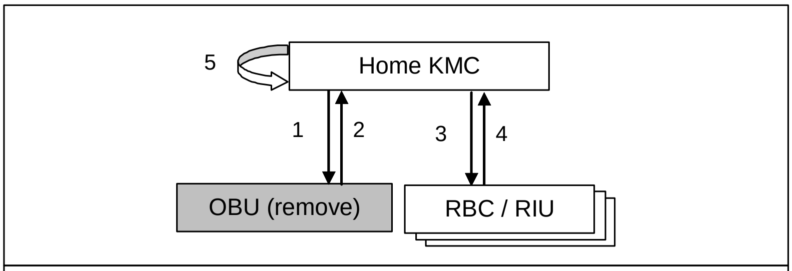
For 1 … number of trackside units (RBC / RIU):

3) "Single" key handling method in RBC/RIU: "Delete Authentication Key"
"All" key handling method in RBC/RIU: "Replace All Authentication Keys"
4) Notify KMC about result

5) Move all keys associated with OBU from key store to archive (e.g. for juridical purposes)

Note 1: the key deletion processes (1 and 3) can be done either in parallel or consecutively.
Note 2: all other OBUs remain unchanged.

A.4.1.2 The next picture illustrates the actions that have to be performed by the KMC when an OBU goes out of commission in its home key management domain. In this example, the KMC used a key management policy that assigns a different authentication key to each OBU – RBC and OBU – RIU relation.



1) "Delete All Keys" from the OBU
2) Notify KMC about result

For 1 … number of trackside units (RBC / RIU):

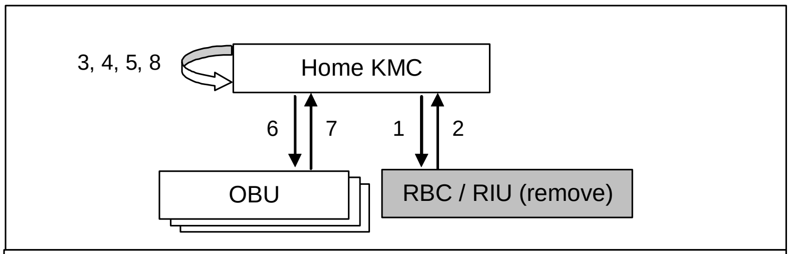
3) "Single" key handling method in RBC/RIU: "Delete Authentication Key"
"All" key handling method in RBC/RIU: "Replace All Authentication Keys"
4) Notify KMC about result

5) Move all keys associated with OBU from key store to archive (e.g. for juridical purposes)

Note 1: the key deletion processes (1 and 3) can be done either in parallel or consecutively.
Note 2: all other OBUs remain unchanged.

A.5 Decommissioning of trackside entity

A.5.1.1 This picture shows the actions that have to be performed by the KMC when trackside entity, e.g. RIU or RBC, goes out of commission in the home key management domain. In this example, the same authentication key has been used to protect the communication of one OBU with all RBCs / RIUs in this domain.



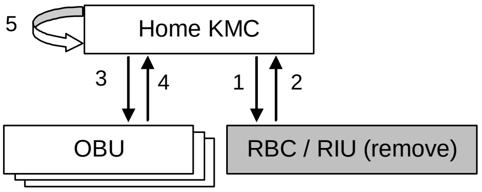
1) “Delete All Keys” from the RBC / RIU that goes out of commission
2) Notify KMC about result

For k = 1 … number of OBUs:

3) Retrieve key related to OBU k from key store
4) Remove identifier of RBC / RIU from trackside list of the key
5) Store updated key in key store if it has not been the last RBC / RIU assigned to the key (trackside list not empty)
6) “Single” key handling method in OBU k:
   “Replace ETCS Entities” of the related the key. If there is no more ETCS entities associated to the key, “Delete Authentication Key”
   “All” key handling method: “Replace All Authentication Keys”
7) Notify KMC about result
8) Move key from key store into archive (e.g. for juridical purposes) if it is the last trackside unit assigned to that key

Note 1: the key deletion processes (1 and 6) can be done either in parallel or consecutively.
Note 2: all other RBCs / RIUs remain unchanged.

A.5.1.2 The next picture illustrates the actions that have to be performed by the KMC when trackside entity, e.g. RIU or RBC, goes out of commission in the home key management domain. In this example, the KMC used a key management policy that assigns a different authentication key to each OBU – RBC and OBU – RIU relation.



1) “Delete All Keys” from the RBC / RIU that goes out of commission
2) Notify KMC about result

For k = 1 … number of OBUs:

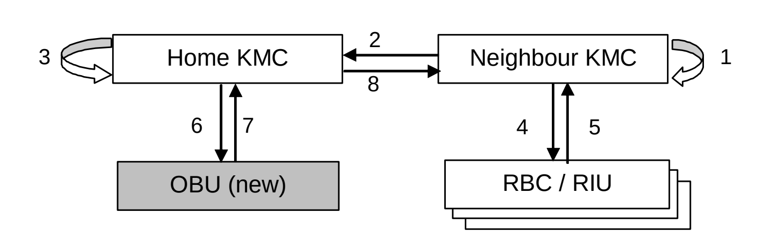
3) “Single” key handling method in OBU k: “Delete Authentication Key”
“All” key handling method in OBU k: “Replace All Authentication Keys”
4) Notify KMC about result
5) Move key associated with this RBC / RIU from key store to archive (e.g. for juridical purposes)

Note 1: the key deletion processes (1 and 3) can be done either in parallel or consecutively.
Note 2: all other RBCs / RIUs remain unchanged.

Annex B (INFORMATIVE) CROSS-DOMAIN SCENARIOS

B.1 Enable a train to cross the domain border

B.1.1.1 This picture shows the actions that have to be performed by the KMCs to enable an OBU to cross the KMC border. In the depicted example, the same authentication key is used to protect the communication of the OBU with all trackside entities (RBC, RIU) of the foreign domain. The neighbourg KMC (Home KMC of trackside entities) generates the key and initiates the key exchange.

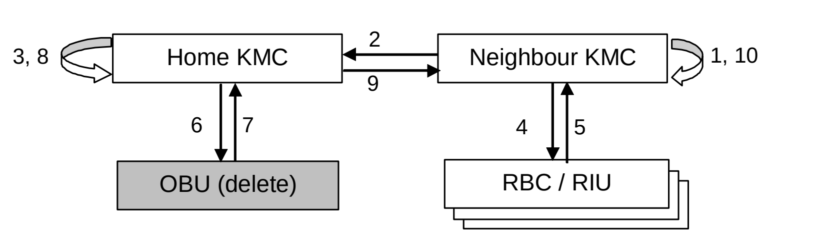


1) Generate a new key and store it in the key store
2) Exchange the new key with the home KMC
3) Store key in the key store
For 1 … number of trackside units (RBCs / RIUs):
4) “Single” key handling method in RBC/RIU: “Add Authentication Key”
“All” key handling method in RBC/RIU: “Replace All Authentication Keys”
5) Notify KMC about result
6) “Single” key / “all” key handling method in new OBU: “Add Authentication Key”/ “Replace All Authentication Keys”
7) Notify KMC about result
8) Inform neighbour KMC about result of performed action

Note 1: the installation (4 and 6) can be done either in parallel or consecutively.
Note 2: all OBUs that are already in service remain unchanged.

B.2 Disable a train to run in a foreign domain

B.2.1.1 This picture shows the actions that have to be performed by the KMCs when an OBU shall be disabled from running in another KMC domain. In the depicted example, the same authentication key is used to protect the communication of the OBU with all trackside entities (RBC, RIU) of the foreign domain. The foreign KMC initiates the key deletion. This is not mandatory - it is also possible that the key deletion is triggered by the Home KMC.

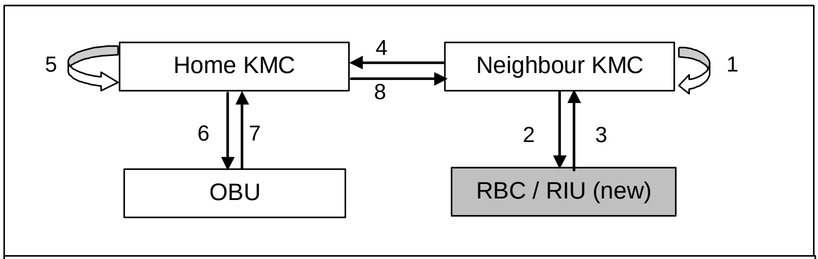


1) Retrieve key from key store
2) Request key deletion from the home KMC
3) Retrieve key from key store
For 1 … number of trackside units:
    4) "Single" key handling method in RBC/RIU: "Delete Authentication Key"
    "All" key handling method in RBC/RIU: "Replace All Authentication Keys"
    5) Notify KMC about result
6) "Single" key handling method in OBU: "Delete Authentication Key"
"All" key handling method in OBU: "Replace All Authentication Keys"
7) Notify KMC about result
8) Move key from key store to archive (e.g. for juridical purposes)
9) Inform neighbour KMC about result of performed action
10) Move key from key store to archive (e.g. for juridical purposes)

Note 1: the deletion processes (4 and 6) can be done either in parallel or consecutively.
Note 2: the deletion request can be triggered either by the home or the neighbouring KMC.
Note 3: all other OBUs remain unchanged.

B.3 Introduction of new trackside entity into service

B.3.1.1 The picture shows the transactions that have to be performed by the KMCs when the new trackside entity is introduced in the foreign domain. In the displayed example, the same authentication key is used to protect the communication of the OBU with all trackside entities (RBC, RIU) of the foreign domain. The foreign KMC initiates the update of the OBU key.

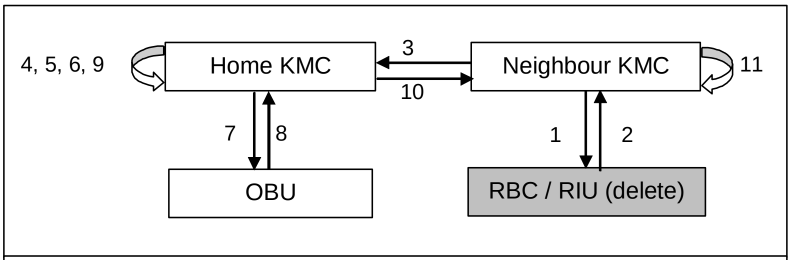


1) Retrieve key from key store, add new trackside identifier and store updated key in key store
2) “Single” key / “all” key handling method in new RBC/RIU: “Add Authentication Key”/ “Replace All Authentication Keys”
3) Notify KMC about result
4) Send update request with new trackside identifier to home KMC
5) Retrieve key from key store, add new trackside identifier and store updated key in key store
6) “Single” key handling method in OBU: “Replace ETCS Entities” of related key installed in the OBU
“All” key handling method in OBU: “Replace All Authentication Keys”
7) Notify KMC about result
8) Inform neighbour KMC about result of performed action

Note 1: the steps 2 and 6 can be done either in parallel or consecutively.
Note 2: all other OBUs and RBCs / RIUs remain unchanged.

B.4 Decommissioning of trackside entity

B.4.1.1 This picture shows the transactions that have to be performed by the KMCs when the trackside entity goes out of commission in the foreign key management domain. In the example, the same authentication key is used to protect the communication of the OBU with all trackside entities (RBC, RIU) of the foreign domain. The foreign KMC initiates the update of the OBU key.



1) “Delete All Keys” from the trackside equipment (RBC / RIU) that goes out of commision
2) Notify KMC about result
3) Send removal request with trackside identifier to home KMC
4) Retrieve key from key store
5) Remove RBC / RIU identifier from trackside list of key
6) Store updated key in key store if trackside list is not empty
7) “Single” key handling method in OBU: “Replace ETCS Entities” of the related the key. If there is no more ETCS entities associated to the key, “Delete Authentication Key”.
“All” key handling method in OBU: “Replace All Authentication Keys”.
8) Notify KMC about result
9) Move key from key store into archive (e.g. for juridical purposes) if it is the last trackside unit assigned to the key
10) Inform neighbour KMC about result of performed action
11) Move all keys related only to the trackside unit that goes out of commission from key store to archive (e.g. for juridical purposes)

Note 1: The steps 1 and 7 can be done either in parallel or consecutively.
Note 2: All other OBUs and RBCs / RIUs remain unchanged.