Requirement specification for EasyGo Basic RSE and OBE

Enclosure F to Document 202
“Roadside and on board equipment”

Document: 202-F
Version: 1.0
Date: 7 December 2018

Table of contents

Document revision history

Version	Date	Author	Main changes
0.1	07.12.2017	TIR	Source document 202-A and B
0.2	12.01.2018	TCL	Included description of testing (Source document 202-E)
0.3	22.01.2018	TCL	Modified due to input WG2-meeting Jan.17-2018
0.4	06.06.2018	TCL	Modified due to input WG2-meeting Apr.24-2018, incl. changed scope for document
0.5	11.06.2018	ASK	Comments
0.6	16.06.2018	TCL	Modified due to ASK's comments
0.7	19.06.2018	TCL	Some modifications
0.8	29.06.2018	TIR	Added BroBizz and PISTA appendices
0.9	09.11.2018	TCL	Some minor modifications based on input
0.95	30.11.2018	ASK	Updated after review on 22 Nov WG2 meeting
1.0	07.12.2018		Approved by ESC

1 Introduction

1.1 General

In order to offer interoperable toll services in EasyGo basic, Toll Service Providers (TSP's) have to issue approved on-board equipment (OBE) to their customers. Toll Chargers (TC's) shall provide Road Side Equipment (RSE) which is able to read all approved OBE.

As requirements for OBE and RSE in an interoperable toll service are mutually dependant, this document is a combined overall requirement specification for both OBE and RSE in the EasyGo basic service. This document describes all aspects of EasyGo basic except testing which is included in document 206 "EasyGo test strategy" and document 207 "EasyGo test specifications".

The following OBE types shall be supported in the EasyGo basic service:

a) EN15509 (EasyGo basic)
b) Pre-15509 PISTA
c) Pre-15509 AutoPASS OBE
d) Pre-15509 BroBizz OBE

OBE of type a) is the current standard EFC application that all new OBEs in EasyGo basic have to comply to. Requirements to this OBE type are described in this document. OBE of type b), c) and d) are all pre-15509 OBEs that no longer will be delivered, but as long as they are valid in EasyGo basic they must be supported by TC's. This document includes descriptions of these OBEs, sufficient for TC's to provide RSE that supports these pre-15509 OBEs.

This document lists some essential requirements as well as recommended solutions for problems that might occur in specific situations or in some EFC systems, along with configuration data.

As it can never be guaranteed that referenced specifications or standards are not conflicting, ambiguous, incomplete or unintentionally leave room for interpretation, the supplier is obliged to contact the purchasing TC and/or TSP for clarification.

It must be noted that EasyGo management in October 2018 has sent a letter to EasyGo TSPs stating that the pre-15509 AutoPASS and BroBizz OBEs should be phased out. The time schedule is:

- From 1 Mar 2019, TC's are no longer required that their RSE read pre-15509 AutoPASS OBEs or pre-15509 BroBizz OBEs
- From 1 Jun 2019, TC's are no longer required to generate video-based transactions (C7 / C8) from such OBEs if they are not read correctly

PISTA OBEs will be phased out later.

1.2 Objectives

This document has two objectives:

1. enable TSP to select suitable CEN DSRC OBEs for operation in EasyGo basic context in EasyGo toll systems
2. enable TC to select suitable CEN DSRC RSE, compliant with OBE provided by TSP's, for operation in EasyGo basic context in EasyGo toll systems

[EasyGo-202] specifies overall principles and requirements for technical interoperability between RSE and OBE in EasyGo, and it covers EasyGo basic as well as EasyGo+. [EasyGo-202] contains several requirements both for RSE and OBE that must be considered in conjunction with this document. In addition, [EasyGo-202-A] contains several requirements and technical descriptions for EasyGo+ OBE of type CEN EN15509 which also apply for EasyGo basic OBEs.

Detailed specifications of pre-15509 OBEs (AutoPASS, BroBizz and PISTA) are included in this document, sufficient for RSE suppliers to support these OBE types.

This document has the objective of going deeper into some issues and only refers to the relevant information in other 202-documents.

1.3 Differences between EasyGo basic and EasyGo+

EasyGo offers two services; - EasyGo basic and EasyGo+. EasyGo+ is specifically designed for vehicles over 3.5 tons driving in Austria and Slovenia. Compared to EasyGo basic the EasyGo+ service has the following additions:

- All EasyGo+ OBEs have the vehicle number plate, nationality and Euro emission category personalised in the OBE
- The number of axles has to be set on the OBE by the driver to indicate the correct number of axles when driving in Austria

Other documents (basically [EasyGo-202-A] to [EasyGo-202-E]) describe EasyGo+ requirements to the current standard EFC applications based on CEN EN15509. EasyGo+ contains basically the same requirements as for EasyGo basic but with some additions. This document therefore refers to other 202-documents where the requirements are the same as for EasyGo+ / EETS. Technically the main differences between EasyGo basic EN15509 and EasyGo+ CEN EN15509 OBE can be summarized in the following:

- Compared to EasyGo+ CEN EN15509, only a subset of attributes is used (or may optionally be used) in EasyGo basic EN15509
- There are different MMI requirements. EN15509 EasyGo basic OBE only has an acoustic element and no MMI for axel selection as EasyGo+ requires.
- While security level 1 is mandatory in EasyGo+, the EasyGo basic OBE of type EN15509 may operate with both security level 0 and level 1

1.4 Core requirements

1.4.1 Requirements for TC

The following OBE types shall be supported by RSE in the EasyGo basic service:

a) EN15509 (EasyGo basic)
b) Pre-15509 PISTA

c) Pre-15509 AutoPASS OBE
d) Pre-15509 BroBizz OBE

In addition, EasyGo+ OBE shall be supported according to EasyGo basic EN15509 requirements.

The RSE interface to EasyGo basic OBEs shall follow the specifications stated in:

- Chapter 2 DSRC for a description of the application interface and basic security principles to be supported
- Chapter 3 User interface for a description of the MMI requirements
- Chapter 4 EETS OBE application element contents for a description of application data elements for all types of OBEs in EasyGo basic
- Chapter 7 APPENDIX A - Security
- Chapter 8 APPENDIX B – Transaction model
- Chapter 9 APPENDIX C - Security principles in AutoPASS
- Chapter **Fejl! Henvisningskilde ikke fundet. Fejl! Henvisningskilde ikke fundet.** gives a detailed description of the implementation of PISTA application, including security principles
- Chapter **Fejl! Henvisningskilde ikke fundet. Fejl! Henvisningskilde ikke fundet.** gives a detailed description of the implementation of the BroBizz application

It should be noted that is voluntary for EasyGo TCs to use security keys and implement security algorithms for pre-15509 AutoPASS.

The RSE interface to EasyGo basic OBE shall provide data sets as described in [EasyGo-202]

The RSE interface to EasyGo basic OBE shall follow general transaction principles and be compliant to content of transaction record as stated in [EasyGo-202]

The RSE shall follow OBE validation principles of OBE as stated in [EasyGo-202]

1.4.2 Requirements for TSP

The OBE shall provide a platform for services and functions available in the framework of a CEN DSRC communication within a vehicle

All new OBEs to be introduced in EasyGo basic shall comply to the standard EFC application EN15509

The OBE must be able to communicate in a multilane environment with overlapping communication zones using different RF-channels. The performance of the OBE must not decrease due to the multilane free-flow functionality.

The EN15509 EasyGo basic OBE shall follow the specification stated in:

- Chapter 2 DSRC gives a description of the application interface and basic security principles to be supported
- Chapter 3 User interface gives a description of the MMI requirements

- Chapter 4 EETS OBE application element contents – gives a description of application data elements for all types of OBE in EasyGo basic. Only parts describing EN15509 EasyGo basic OBE are applicable.
- Chapter 7 APPENDIX A - Security gives a general description of security features. Only parts describing EN15509 EasyGo basic OBE are applicable.

2 DSRC

2.1 DSRC Interface

The OBE and RSE support DSRC communications at 5,8 GHz and must conform to **ISO 14906**.

2.2 Application

The OBE and RSE shall support any DSRC post pay transaction whose attributes, parameters, functions and security features are according to this document.

The transaction is permitting data exchange for tolling via the DSRC interface.

2.2.1 Application interface for EFC

The table below specifies the EFC functions that are supported according to [L7] and EFC API] as actions:

Name	Action Type	Action Parameter	Response Parameter	Remarks
Get_Stamped	0	GetStampedRq	GetStampedRs	Retrieves data with an authenticator from the OBE
Get_Nonce	6	-	Octet String	Reads a random number generates by OBE Optional.
Set_MMI	10	SetMMIRq	-	Invokes an MMI function (e.g. signal OK via buzzer). Used **only for EN15509 OBE protocol**
Echo	15	Octet String	Octet String	OBE echoes received data
Get_Secure	3	GetSecureRq	Get_Secure response	Used **only for pre-15509 AutoPASS OBE protocol** to retrieve data

Table 1 Action functions

2.2.2 Data elements

Data elements are defined in chapter 4 in this document.

For tariff calculation, different information (different attributes) is needed in the different systems of EFC operators. Data to be stored into the OBE's attributes is defined in this document.

2.3 Security

For a description of the security architecture for EN15509, Pre-15509 AutoPASS, PISTA and BroBizz, refer to chapter 7 APPENDIX A - Security.

All security requirements stated in [EasyGo-202-A] chapter 2.3 are also applicable for EN15509 EasyGo basic OBE.

2.4 Additional requirements and remarks

All requirements stated in [EasyGo-202-A] chapter 2.4 related to:

- Multilane free-flow ability
- Slow response
- SET_MMI.request command
- Data storage
- Multiple transaction

…are also applicable for EN15509 EasyGo basic OBE.

Similar requirements related to:

- Multilane free-flow ability
- Data storage
- Multiple transaction

…are also applicable for pre-15509 AutoPass, PISTA and BroBizz

3 User interface

The driver might be informed about the status of the toll transaction after passing a tolling station as defined by the TSP. This information may be achieved by a MMI on the OBE.

Of the OBE supported by EasyGo basic (EN15509, pre-15509 AutoPASS, PISTA and BroBizz) only EN15509 has MMI.

There is however no MMI supported in the new RSE in Norway.

3.1 MMI elements

The EasyGo basic EN15509 OBE shall contain a buzzer with one tone as an acoustic information element. The buzzer shall be able to signalize the SET-MMI-Codes 0 to 2 and 255 according to the table below representing the buzzer signalization for the transaction when passing a RSE:

Transaction result	SET-MMI-Code	Buzzer
Transaction OK (payment done, no warning)	0	1 short beep
Transaction not OK (no payment effected, for example, due to expired contract)	1	4 short beeps
Warning (use is TC specific)	2	2 short beeps
Particular scope or future use	255	No beep

Table 2 Signalization when passing an RSE

The RSE must be able to control the activation of the buzzer by a parameter. The use of a buzzer is optional for RSE in EasyGo basic.

3.2 Instructions to users

The TSP shall provide an electronic or printed instruction / description containing the description of the OBE user interface functionalities and the reference to the local tolling regulations of the EasyGo toll domains concerned.

The TC shall describe their requirements to OBE in order to provide users with correct information of status when passing through a toll station. TC and TSP must agree on how this information is given to the user. An alternative to audible signal is a signal light installed at roadside.

4 EETS OBE application element contents

4.1 Application elements for EasyGo basic EN15509 OBE

The table below shows an overview of the application data elements for an EN 15509 EasyGo basic OBE, for the detailed description of the key elements EFC Context Mark and Payment means used in EasyGo basic see next chapter.

Attributes (EID>0)	AttrId	Type	Length in Bytes	Read	Write	Remarks
CONTRACT						Information associated with the service rights of the Contract Provider
EFC Context Mark	0	32	6	Yes	No	Contains the Contract Provider Identification. Transmitted as part of the VST.
PAYMENT						Data associated with the Payment transaction.
PaymentMeans (including PAN)	32	64	14	Yes	No	Includes: - The Personal Account Number, including the Payment Means Issuer (identified by the IIN), - The PAN Expiry Date - The payment means Usage Control
VEHICLE						Information pertaining to the identification and characteristics of the vehicle.
VehicleLicencePlateNumber	16	47	Variable 13 to 17 bytes	Yes	No	Length of the attribute, incl. Country code, Alphabet Indicator and length. *) ***Optional***
VehicleClass	17	49	1	Yes	No	***Optional***
VehicleDimensions	18	50	3	Yes	No	***Optional***
VehicleAxles	19	51	2	Yes	No	***Optional***
VehicleWeightLimits	20	52	6	Yes	No	***Optional***
VehicleSpecificCharacteristics	22	54	4	Yes	No	***Optional***
EQUIPMENT						Information pertaining to the OBE.
EquipmentOBEId	24	56	5 (=4+1)	Yes	No	***Optional***
EquipmentStatus	26	58	2	Yes	Yes	***Optional***
RECEIPT						
ReceiptData1 (last)	33	65	28	Yes	Yes	***Optional***
ReceiptData2 (penultimate)	34	66	28	Yes	Yes	***Optional***

Table 2 EETS OBE application element contents

Implementation of additional attributes for compatibility reasons to other existing systems (like AttrID. 4 and 23) is up to the TSP.

*) According to EN15509 the length of attribute 16 VehicleLicencePlateNumber can be (10 to 14) + 3 bytes. Though the RSE can read LPN information with a length of up to 14 characters, only the first 10 significant characters are further processed in the central systems of EasyGo TC's.

"Read" and "Write" define access rights to a given attribute for GET, GET_STAMPED or SET used by RSE.

Each attribute contains one or several data fields. Personalization shall be made by the Contract Issuer (TSP) as specified in [EasyGo-202-B], which is based on [EFC API].

4.2 Application elements for all types of OBE in EasyGo basic

The table below shows an overview of the application data elements, used by any of the EFC Applications in EasyGo Basic.

Attribute ID			**EFC Application**			
Attribute name	**Id**		**EN15509**	**PISTA**	**BroBizz**	**AutoPASS**
Contract *(Information associated with the service rights of the TSP of the EFC service)*						
EFC Context Mark	0		Yes	Yes	Yes	Yes
Contract Authenticator	4			Yes		
Vehcicle *(Identification and characteristics of the vehicle.)*						
Vehicle Licence Plate No	16		Yes			
Vehicle Class	17		Yes	Yes		
Vehicle Axles	19		Yes	Yes		
Vehicle Specific Characteristics	22		Yes			
Equipment *(Identification of the OBU and general status information*						
Equipment OBU ID	24		Yes	Yes		
Equipment Status	26			Yes		
Payment *(Data identifying the Payment means and its validity)*						
Payment Means	32		Yes	Yes		
NTD Data 1	99					Yes
PAN/OBU ID	101				Yes	
Receipt *(Financial and operational information associated with a specific session.)*						
Receipt Data 1	33			Yes		
Receipt Data 2	34		Yes	Yes		
Other data						
Private Licence plate number	91			Yes		

Attribute ID			EFC Application			
Attribute name	**Id**		**EN15509**	**PISTA**	**BroBizz**	**AutoPASS**
Private shadow class	92			Yes		
Private reserve	93			Yes		
Private Blacklist	94			Yes		
NTD Data 2	100					Yes
NTD Data 3	127					Yes

Table 3 EasyGo Basic OBE application element contents

5 Attribute data

The general structure of Attributes and their data elements is shown in the previous chapter. Only the following attributes are mandatory for tolling purposes in EasyGo basic:

- Attribute 0: EFC-Context Mark
- Attribute 32: PaymentMeans (only for EN15509, pre-15509 PISTA)
- Attribute 99: NTD Data1 (only for pre-15509 AutoPASS)
- Attribute 101: TRP ID (only for pre-15509 BroBizz)

For a description of all other attributes that are optional in EasyGo basic, reference is made to [EasyGo-202-B] and table 4 in section 4.2.

5.1 Attribute 0: EFC-Context Mark

The **EFC-ContextMark** denotes a specific EFC context in the OBE, comprising the organisation that issued the contract, the type of contract and the context version. EFC-ContextMark data is transmitted in VST as part of the ApplicationContextMark to enable the RSU to select the suitable EFC application as well the appropriate OBE data element, if the OBE is presenting more data elements. For more information see Attribute 0 in [EasyGo-202-B].

5.2 Attribute 32: PaymentMeans

The attribute PaymentMeans holds the contract data as PAN, ExpiryDate and UsageControl for EN15509 and pre-15509 PISTA:

Data element	Definition	Use in EasyGo context
PersonalAccount Number (PAN)	Coded according to financial institutions, consists of the Major Industry Identifier (MII), the Issuer Identifier Number (IIN, including the MII), the account number and a check digit (calculated with the Luhn algorithm).; acc. to ISO7812	Mandatory
PaymentMeans ExpiryDate	Expiring date of payment means. Payment means expires at 24h of PaymentMeans ExpiryDate. (Expiry date to be choosen by issuing TSP)	Mandatory *)
PaymentMeans UsageControl	Indicates issuer's specified restrictions on the geographic usage and services allowed for the applications	Not used, optional (or set to zero)

Table 5 Payment means

*) Note: Interpretation of PaymentMeansExpiryDate is mandatory at RSE, so the OBE must be personalized with a date freely chosen by the issuing TSP.

5.3 Attribute 99: NTD Data 1 (PAN/OBU ID)

The attribute NTD Data 1 holds the data of PAN/OBE ID in pre-15509 AutoPASS. See Chapter 8.2 Pre-15509 AutoPASS.

5.4 Attribute 101: PAN/OBU ID (TRP ID)

The attribute TRP ID holds the data of PAN in pre-15509 BroBizz:

The value of the TRP ID shall have the following format

For the BroBizz application the number of BCD-coded digits of the TRP ID shall be fixed to 9. This means that the length of the TRP ID will be 8 bytes.

The first 3 bytes correspond to the first 3 bytes of the AFC Context Mark. The example below contains the Issuer ID of BroBizz. Øresund has an Issuer ID with the digits 46001x coded in the PAN/OBU ID.

Issuer-ID						Individual Account Identification									Check digit
9	7	8	0	0	3	BCD	BCD	BCD	BCD	BCD	BCD	BCD	BCD	BCD	BCD

Table 6 The TRP ID

6 Comments on DSRC protocol related issues (informative)

These requirements apply for EN15509 only and provide information on some DSRC protocol related issues often raised by OBE manufacturers and information about tolling context specific characteristics.

All requirements stated in [EasyGo-202-B] chapter 7 related to:

- Comments on OBE DSRC Kernel state after Rec_PrWA event
- Comments on Sleep after release
- Comments on OBE Random number generation Data storage

… are also applicable for EN15509 EasyGo basic OBE.

7 APPENDIX A - Security

7.1 General

The different EFC Applications have different security architectures which are briefly described in the following subchapters.

EN15509 with security level 1 uses access credentials, while neither pre-15509 AutoPASS, BroBizz nor PISTA use access credentials.

7.2 Description of security architecture in EN15509

The European Standard EN 15509 defines security features and mechanisms based on the general security framework defined in EN ISO 14906. EN 15509 allows for implementation of two different security levels (0, 1). Security level 0 is mandatory while security level 1 is optional. Currently new AutoPASS OBEs based on EN 15509 use security level 1 (EasyGo+ OBEs also has this security level). Other EN 15509 OBEs in EasyGo basic use security level 0.

Security level 0 defines calculation of an authenticator to validate data integrity and origin of application data. A Message Authentication Code (MAC) is calculated using a DEA algorithm according to ANSI X3.92

- For EasyGo the key reference range for TSP Authenticator (CI authenticator, KeyRef to be used in the first GET_STAMPED.request) is 111 to 114.
- For EasyGo the key reference range for TC Authenticator (Operator authenticator, KeyRef to be used in the second GET_STAMPED.request) is 115 to 118.

Keys are calculated according to EN 15509 [IAP].

For EasyGo basic OBEs that use security level 1, the use of Access Credentials is mandatory.

The EN15509 OBE can store at least:

- eight AuthenticationKeys
- one AccessKey

All keys stored in the OBE shall be protected against read out. There shall be no read access to authentication keys as well as to access keys.

7.2.1 Authentication

Authentication is obtained by the GET_STAMPED command.

In AutoPASS, also the GET_SECURE command is used.

The RSE requires authentication from the OBE. For this purpose, the GET_STAMPED function is used with 2 different authenticator keys (operator and issuer authenticator). Therefore, the OBE shall authenticate the requested data.

For details see also [IAP].

7.2.2 Speed of security calculations

In order to support free-flow systems the OBE shall execute security calculations with sufficient speed such that the transaction duration is successfully completed in less than 70ms. The transaction duration is measured in the communication zone of any free-flow RSE from the first BST message to the reception of a RELEASE or ECHO message.

7.2.3 Protection of code, data and keys

If the architecture of the OBE allows in principle readout of code or data e.g. for debugging purposes on test OBEs, this must not be possible for units delivered for customer use.

All keys stored in the OBE shall be protected against unauthorised read out. There shall be no read access to AuthenticationKeys as well to AccessKeys.

7.3 Description of security architecture in pre-15509 PISTA

The PISTA security architecture is based on two different and complementary levels of security, named respectively Data Certification and OBE Authentication. No access credentials mechanism is part of the common service. Each OBE supports both levels of security, and the RSE is responsible for selecting the one to be applied by means of relevant function invocation. The OBE shall be fully initialised since the beginning and shall store all keys even if they shall not be used from the beginning. The common service shall be based upon the first level of security, Data Certification, and then eventually upgraded to higher levels of security, OBE Authentication, in case a significant level of fraud is detected.

The security architecture is described further in appendix E.

7.4 Description of security architecture in pre-15509 BroBizz

The security architecture is described in appendix E.

7.5 Description of security architecture in pre-15509 AutoPASS protocol

The AutoPASS security scheme is based upon the following principle:

- two (2) types of keys are stored into the OBE, and for each type of keys, five (5) generations of cryptographic keys are stored in the OBE.
- two (2) Message Authentication Codes (MACs) are computed and returned by the OBE. Checking these MACs allows the TCs to take any appropriate actions in case the OBE is not authentic (i.e. a photo of the licence plate of the violating vehicle).

The first Message Authentication Code (MAC1) is used by a TC to check the authenticity of an OBE that the “native” TSP has issued (combined TC/TSP role common in Norway, but it will be separated in the future). The second Message Authentication Code (MAC2) is used for interoperability purposes, to allow another TC/TSP than the one who issued the OBE to perform an early detection of a foreign vehicle equipped with an unauthorised/illegal/illicit OBE.

No access credentials mechanism is part of the pre-15509 AutoPASS.

See appendix C for a detailed description of the security principles in pre-15509 AutoPASS.

8 APPENDIX B – Transaction models

8.1 EN15509

See EN14906 ([EFC API]) chapter B.4.3.1 and B.4.3.2.

8.2 Pre-15509 AutoPASS

Octet #	Attribute / Field		b7 b6 b5 b4 b3 b2 b1 b0	Description
0	Private LID		x x x x x x x 0	Link address of a specific OBU
1			x x x x x x x 0	
2			x x x x x x x 0	
3			x x x x x x x 1	
4	MAC control field. L		1	The frame contains a LPDU
	MAC control field. D		0	Direction is down link
	MAC control field. A		1	Private Up Link Window is allocated
	MAC control field. C/R		0	Command LPDU
	MAC control field. S		s	Sequence bit. At RSU the S bit is set equal to V(A). At the OBU V(A) is set equal to S.
	MAC control field. reserved bits		0 0 0	Reserved.
5	LLC control field. n		n	ACn command frame bit
	LLC control field. M		1 1 0 1	ACn command
	LLC control field. P/F		1	Polling
	LLC control field. reserved bits		1 1	Not used. Always set to 1.
6	Fragmentation header		1 0 0 1 0 0 0 1	No fragmentation
7			0 0 0 0	ACTION.request
	GET_SECURE.request	SEQUENCE		
	{			
	OPTION indicator		1	AccessCredentials present
	OPTION indicator		1	ActionParameter present
	OPTION indicator		0	IID not present
	Mode	BOOLEAN	1	Reply expected
8	EID	INTEGER(0..127,...)	0 0 0 0 0 0 0 1	No extension, EID = 1
9	ActionType	INTEGER(0..127,...)	0 0 0 0 0 0 1 0	No extension, GET_SECURE.request = 2
10	AccessCredentials	OCTET STRING	0 0 0 0 1 0 0 1	No extension, string length = 9 octets
11	{			
	KeyGeneration	INTEGER(0..127,...)	0 0 0 0 0 g g g	Key generation.
12	RND-1	OCTET STRING(SIZE(4)	r r r r r r r r	(LSB) 4 octets random number challenge from roadside
13			r r r r r r r r	
14			r r r r r r r r	
15			r r r r r r r r	
16	TVP	OCTET STRING(SIZE(4)	t t t t t t t t	(LSB)
17			t t t t t t t t	4 octets time variable parameter used in encryption
18			t t t t t t t t	
19			t t t t t t t t	
20	}			
	ActionParameter	CONTAINER	0 0 0 0 0 0 1 0	No extension, Type 2 = Octet string
21			0 0 0 0 0 0 1 0	No extension, octet string length = 2
22	{		0 0 0 0 0 0 0 1	No extension, number of attributes = 1
23	First attributeId	INTEGER(0..127,...)	0 1 1 0 0 0 1 1	Attribute AutoPASSdata1; attributeId=99.
	} }			

Table 7 GET-SECURE.REQUEST

Octet #	Attribute / Field		b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	Description
0	Private LID		x x x x x x x **0**	Link address of a specific OBU
1			x x x x x x x **0**	
2			x x x x x x x **0**	
3			x x x x x x x **1**	
4	MAC control field. L		1	The frame contains a LPDU
	MAC control field. D		1	Direction is up link
	MAC control field. R		0	Private Up Link Window is not requested
	MAC control field. C/R		1	Response LPDU
	MAC control field. reserved bits		0 0 0 0	Reserved.
5	LLC control field. n		n	ACn command frame bit
	LLC control field. M		1 1 0 1	ACn command
	LLC control field. P/F		1	Final bit set.
	LLC control field. reserved bits		1 1	Not used. Always set to 1.
6	LLC status field. RRRR		0 0 0 0	Response available
	LLC status field. CCCC		0 0 0 0	Command accepted
7	Fragmentation header		1 0 0 1 0 0 0 1	No fragmentation
8			0 0 0 1	ACTION.response
	GET_SECURE.response	SEQUENCE		
	{			
	OPTION indicator		0	IID not present
	OPTION indicator		1	ResponseParameter present
	OPTION indicator		0	ReturnStatus not present
	fill	BIT STRING (SIZE(1))	0	
9	EID	INTEGER(0..127,...)	**0** 0 0 0 0 0 0 1	No extension, EID = 1
10	ResponseParameter	CONTAINER	**0** 0 0 0 0 0 1 0	No extension. OCTET String = 2.
11			**0** 0 0 1 1 1 0 0	No extension, octet string length = 28
12	{			
	Attributes	SEQUENCE	**0** 0 0 0 0 0 0 1	No extension. 1 attribute in list
13	{			
	AttributeId	INTEGER(0..127,...)	**0** 1 1 0 0 0 1 1	No extension. AttributeId = 99 (AutoPASSdata1)
14	Attribute Value	CONTAINER	0 0 0 0 0 0 1 0	OCTET STRING = 2
15			0 0 0 1 1 0 0 0	No extension. String length = 24 octets
16	{			
	obuID	CS1	c c c c c c c c	country code c
17			c c	
			i i i i i i	IssuerIdentifier i
18			i i i i i i i i	
19			s s s s s s s s	ServiceNumber s
20			s s s s s s s s	
21			s s s s s s s s	
22			s s s s s s s s	
23	efcStatus	BIT STRING(SIZE(16))	m	Moved bit: m=1: OBU moved, else m=0.
			0	not used for AutoPASS
			b	Battery low bit: b=1: voltage low, else b=0.
			0	not used for AutoPASS
			0	
			0	
			0	
			0	
24			0 0 0 0 0 0 0 0	
25	TC	OCTET STRING(SIZE(2))	t t t t t t t t	(LSB): Transaction counter
26			t t t t t t t t	
27	RND-2	OCTET STRING(SIZE(4))	r r r r r r r r	(LSB): Random number challenge
28			r r r r r r r r	
29			r r r r r r r r	
30			r r r r r r r r	
31	MAC1	OCTET STRING(SIZE(4))	m m m m m m m m	(LSB): Message authenticator
32			m m m m m m m m	
33			m m m m m m m m	
34			m m m m m m m m	
35	MAC2	OCTET STRING(SIZE(4))	n n n n n n n n	(LSB): Message authenticator
36			n n n n n n n n	
37			n n n n n n n n	
38			n n n n n n n n	
39	LogIndex	INTEGER(0..255)	j j j j j j j j	Points to most recent entry in log file
	} } } }			

Table 8 GET-SECURE.RESPONSE

8.3 Pre-15509 PISTA

See EN15509 as PISTA uses the same element.

8.4 Pre-15509 BroBizz

BroBizz uses TRP ID (element 10) a 4-octet string element instead of PAN (element 32) a 14-octet string element. The calculations are described in chapter 11.2.6.3.

9 APPENDIX C - Security principles in AutoPASS

9.1 General

9.1.1 Reliability of the communication channel

The security of the data transmitted via the DSRC interface is provided on an end-to-end basis (application layer to application layer). Issues associated with the reliability of the communication channel are therefore not considered, and consequently these specifications assume the DSRC link to be error-free.

9.1.2 OBE manufacturer's responsibility

Based on the general procurement philosophy, and the security requirements defined in chapter **Fejl! Henvisningskilde ikke fundet.**, the OBU manufacturer is responsible for

- the generation of the keys,
- the secure loading of the keys into the OBE (initialisation/personalisation process)

  NOTE: As a result of the OBE personalisation/initialisation process by the OBE manufacturer, the built-in transaction counter shall be initialised to the value '00 00'
- the secure transmission of the keys to the TC's RSE and CS sub-systems

9.1.3 Security needs and cryptographic keys in the OBU

The present specification answers the following basic security needs:

- enable a RSE and a CS belonging to a TC to check the OBE ID and other application data transmitted by an OBU issued by the same operator (case of a transaction between a native vehicle/OBU and a native RSE)
- enable a RSE (and optionally a CS) belonging to a TC to check the OBU ID and other application data transmitted by an OBU issued by a different operator (case of a transaction between a native RSE and a foreign vehicle/OBU)

In order to make that possible,

- two (2) types of keys are stored into the OBU, and for each type of keys, five (5) generations of cryptographic keys are stored in the OBU.
- two (2) message authentication codes are computed and returned by the OBU. Checking these MACs allows the operators to take any appropriate actions in case the OBU is not authentic (i.e. a photo of the licence plate of the violating vehicle).

The first message authentication code (MAC1) is used by an operator to check the authenticity of an OBU (verify that it is genuine) that he has issued.

The second message authentication code (MAC2) is used by an operator to check the authenticity of the OBU (verify that it is genuine) that other operators have issued.

It should be noted that MAC2 is used for interoperability purposes, to allow another operator than the one who issued the OBU to perform an early detection of a foreign vehicle equipped with an unauthorised/illegal/illicit OBU.

9.1.4 OBU identification and integrity of the data transmitted by the OBU

The identification of the OBU for charging purposes is primarily based on the identification number (OBU ID) attached to each OBU manufactured and stored in the OBU. This OBU ID is returned by the OBU upon request by the RSE. A challenge-response mechanism, and Message Authentication Codes (MAC1 and MAC2) attached to the data transmitted by the OBU are used to ensure the integrity of the OBU ID and counter the associated threats (i.e. impersonation of the OBU, alteration of the OBU ID, replays of transactions, disputes). Associated cryptographic keys are stored in the OBU to enable the computation of the MACs.

9.2 Initial assumptions and transaction scenarios

Given that the following entities and sub-systems are involved in the ETC transactions

- *An Issuer (operator N) represented by*
  1) *the secret keys MKEY-N and MKEY-F*
  2) *on-board units which contains the secret keys OBUKEY-N(i), and OBUKEY-F(i)*
  3) *a roadside equipment RSE-N, which contains the secret key MKEY-F*
  4) *a central system CS-N, which contains the secret key MKEY-N, and which may or may not contain MKEY-F (Note 1)*
- *A vehicle V equipped with an on-board unit OBU-N(i) issued by the Issuer (operator N)*
- *An EFC operator (operator F) represented by*
  1) *a roadside equipment RSE-F which, in addition to its own set of secret keys (optionally), contains also the secret key MKEY-F belonging to the Issuer (operator N)*
  2) *a central system CS-F which may or may not contain MKEY-F (Note 2)*

  NOTE 1: The availability of MKEY-F in CS-N is not required as the key is intended for use by the foreign service provider SP-F.

  NOTE 2: The availability of MKEY-F in CS-F is not required as the key is intended for use by RSE-F.

the following transaction scenarios can be distinguished:

Case 1 - Native vehicle and native EFC operator, i.e. the Issuer and EFC operator is the same entity, e.g. the same toll company:
EFC Transaction between OBU-N(i) and RSE-N

Case 2 - Native vehicle and foreign EFC operator, i.e. the Issuer and EFC operator are different entities, e.g. two different toll companies:
EFC Transaction between OBU-N(i) and RSE-F

9.3 Authentication mechanisms

Case 1 – Transaction scenario involving a native vehicle and a native EFC operator (OBU-N(i) and RSE-N):

In response to the command Get-Secure transmitted by RSE-N, the OBU-N(i) returns the following message:

M = Transaction Data || MAC1 || MAC2

where

- Transaction data itself is the result of the concatenation of the OBU ID, the OBU status, the value of the transaction counter, and the random number generated internally by the OBU
- MAC1 is generated using a session key based on the OBU specific secret key OBUKEY-N(i), the time parameter contained in the BST, and the random number provided by the RSE
- NOTE: MAC1 cannot be checked by RSE-N and therefore is of no relevance for RSE-N.
- MAC2 is generated using a session key based on the OBU specific secret key OBUKEY-F(i), the time parameter contained in the BST, and the random number provided by the RSE
- NOTE: This message authentication code is used by RSE-N to check the validity (authenticity) of the data returned by the OBU (and therefore verify that the OBU is genuine) at the time of transaction

The roadside equipment RSE-N performs the following operations:

1. computing a reference MAC designated as MAC2', using the data returned by the OBU, and OBUKEY-F(i) derived from MKEY-F
2. check the authenticity of the on-board unit OBU-N(i) by comparing MAC2 generated by the OBU and MAC2'
3. initiate/trigger a violation operation if the checking fails or otherwise store the entire transaction details for communication to the central system.
4. transmit the transaction data (including the certificate MAC1 and MAC2) to the central system CS-N

Case 2 – Transaction scenario involving a Native vehicle and a foreign EFC operator (OBU-N(j) and RSE-F):

In response to the Get-Secure transmitted by RSE-F, OBU-N(i) generates the following message:

M = Transaction Data || MAC1 || MAC2

where

- Transaction data itself is the result of the concatenation of the OBU ID, the OBU status, the value of the transaction counter, and the random number generated internally by the OBU
- MAC1 is generated using a session key based on the OBU specific secret key OBUKEY-N(i), the time parameter contained in the BST, and the random number provided by the RSE

  NOTE: MAC1 cannot be checked by RSE-F and therefore is of no relevance for RSE-F.

- MAC2 is generated using a session key based on the OBU specific secret key OBUKEY-F(i), the time parameter contained in the BST, and the random number provided by the RSE

  NOTE: MAC2 is intended to be checked by RSE-F

The roadside equipment RSE-F performs the following operations:

1. computing a reference MAC designated as MAC2', using the data returned by the OBU, and OBUKEY-F(i) derived from MKEY-F
2. check the authenticity of the on-board unit OBU-N(i) by comparing MAC2 generated by the OBU and MAC2'
3. initiate/trigger a violation operation if the checking fails or otherwise store the entire transaction details for communication to the central system.
4. transmit the transaction data (including the certificate MAC1 and MAC2) to the central system CS-F

9.3.1 Disputes and related policy

In case of dispute between Issuer (operator N) and EFC operator (successful check of MAC1 and failed check of MAC2) contractual agreement should stipulate that the final proof is constituted by the MAC1. The rationale for that being that in addition of being stored in the OBU, the key required to compute MAC1 is only in possession of the native operator who is also directly responsible for the OBU.

9.4 Key hierarchy

The cryptographic keys stored in the OBU are generated and used according to the general 3-level hierarchy shown in the figure below.

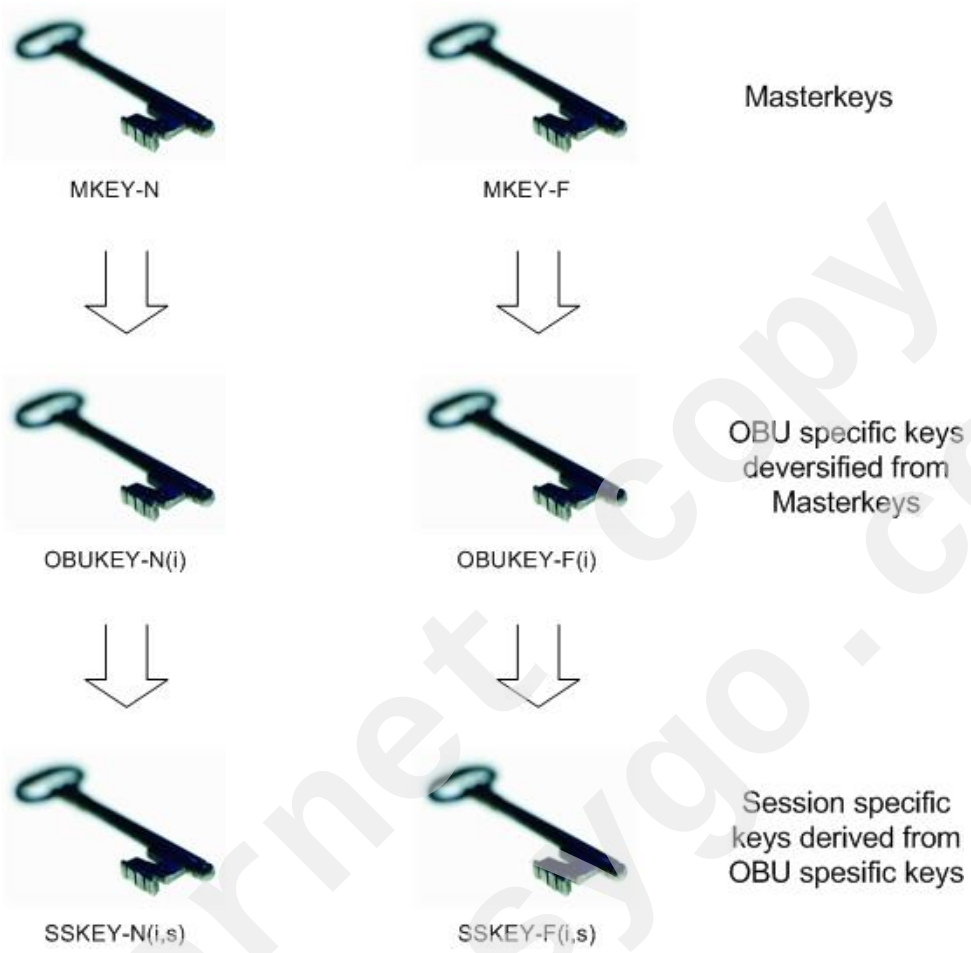


Figure 1: Key hierarchy for the OBU cryptographic keys

At the top level is the master key, denoted *MKEY-N and MKEY-F*. Each master key *MKEY* is a single key, unique for each generation of keys and for each operator.

Down in the next level, are the OBU specific keys *OBUKEY(i)*, denoted *OBUKEY-N(i) and OBUKEY-F(i)*, one for each OBU belonging to the Operator. These keys are obtained by diversifying the corresponding master key *MKEY* with the OBU ID.

The lowest level keys are the session keys *SSKEY(i,s)*, denoted *SSKEY-N(i,s) and SSKEY-F(i,s)* which are transaction specific and are derived from the OBU specific key, and time variant information.

Note: The same key hierarchy is used in relation with the OBU operations in the RSE and the CS.

9.5 Generating the secret keys

9.5.1 Generating the master keys

A set of master keys, specific to a given operator, and comprising two (2) type of keys and five (5) generations for each type of keys, is generated by the OBU manufacturer and used to derive the OBU specific keys. Sub-sets of the master keys are made available to the other sub-systems as described in this document.

The procedures and mechanisms used by the OBU manufacturer to generate the master keys and transmit them to the other sub-systems fall outside the scope of these specifications and therefore are not described in this document.

The procedures and mechanisms used to load the master keys into the other sub-systems fall outside the scope of these specifications and therefore are not described in this document.

Of the three types of master keys, only one type is shared across operators for interoperability purposes. The procedures and mechanisms used to "publish" this particular subset of master keys fall outside the scope of these specifications and therefore are not described in this document.

9.5.2 Generating the OBU specific keys

Each type of OBU specific key OBUKEY(i) is derived from the corresponding master key MKEY for the given using the OBU ID (OBU Identification number). The OBU manufacturer stores this key into the OBU using an appropriate procedure, not described in this document.

9.6 Computing the Message Authentication Codes

The Message Authentication Codes MAC1 and MAC2 are computed by the OBU, using session keys derived from the OBU specific keys stored in the OBU.

It should be noted that the availability of the following data:

- Key Generation number,
- Time parameter contained in the BST,
- RND-1 (Random number generated by the RSE)
- OBU ID,
- OBU status, and
- RND-2 (Random number generated by the OBU)

…in PLAIN TEXT (not encrypted), while MAC1 and MAC2 are the result of an encryption mechanism (MAC operation) makes possible to build interoperable systems gradually, meaning that the security features represented by MAC1 and MAC2 are put in use progressively in the RSE and CS of the operators concerned.

9.7 Data exchanges

The integrity of the data returned by the OBU upon request by the RSE is protected by Message Authentication Codes computed internally by the OBU, using session keys based on OBU specific keys stored in the OBU and time variant parameters.

The security related data exchange between the RSE and the OBU is described in the figure below.

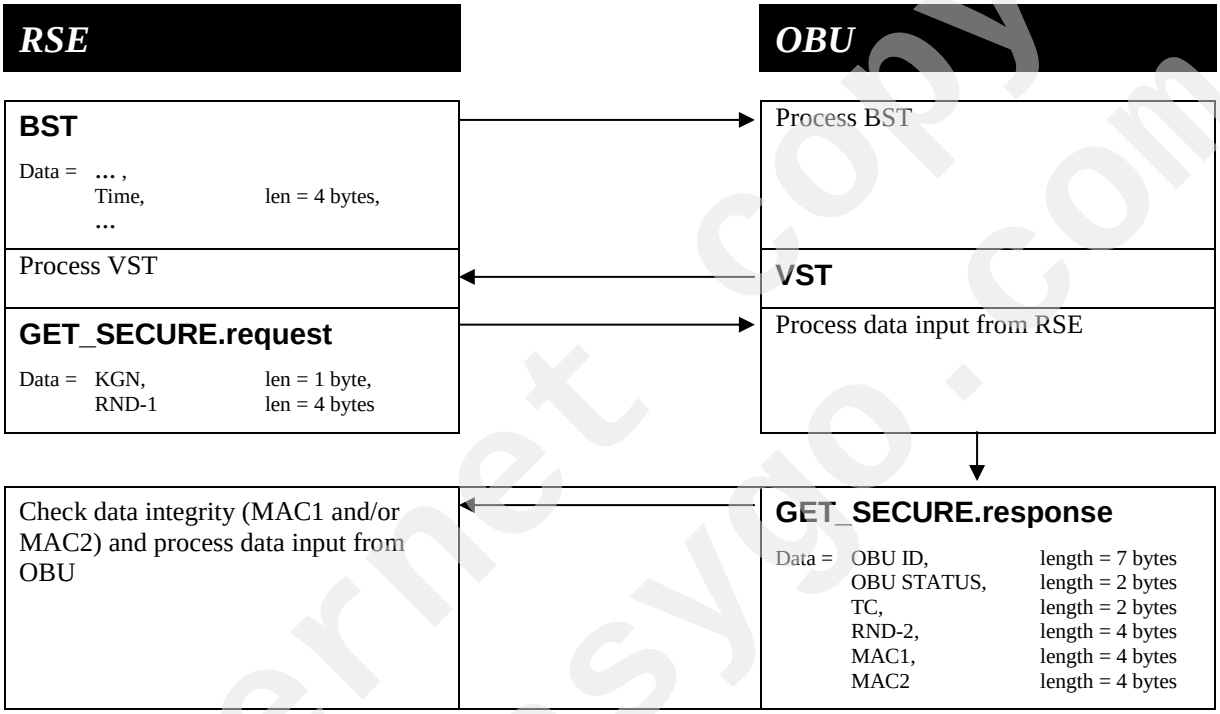


Figure 2: Sequence of data exchange and operations

9.8 Conventions for the DES algorithm and related keys

DES [7] is the cryptographic algorithm used in the generation of the Message Authentication Codes computed by the OBUs and checked by the RSEs:

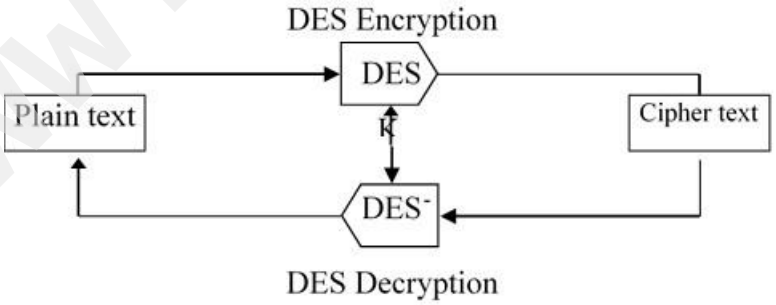


Figure 3: Encryption and decryption using DES

Assuming that *P = Plain Text*, *C = Cipher Text*, *E = DES encryption*, *D = DES decryption*, and *Ks = DES key*, DES encryption and decryption can be expressed as follows:

$$C = E_{Ks}(P) \qquad P = D_{Ks}(C)$$

9.9 Conventions for the triple-DES algorithm and related keys

The cryptographic algorithm to be used for the generation of the OBU specific keys is triple-DES [10], described in the figure below:

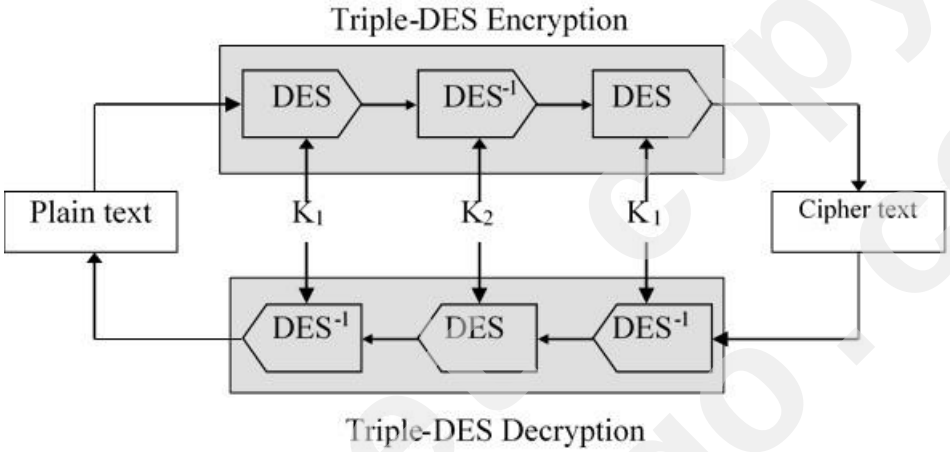


Figure 4: Encryption and decryption using triple-DES

Assuming that *P = Plain Text*, *C = Cipher Text*, *E = DES encryption*, *D = DES decryption*, K_1 *= DES key*, and K_2 *= DES key*, triple-DES encryption and decryption can be expressed as follows:

$$C = E_{K_1}(D_{K_2}(E_{K_1}(P)))$$

$$P = D_{K_1}(E_{K_2}(D_{K_1}(C)))$$

Assuming that *K = triple-DES key as defined below*, that $ede_K()$ *= triple-DES encryption function using a key K*, and that $ded_K()$ *= triple-DES decryption function using a key K*, triple-DES encryption and decryption can also be expressed in the simpler form, which will be used throughout the remainder of the document:

$$C = ede_K(P)$$

$$P = ded_K(C)$$

Conventions for triple-DES keys

Each triple-DES key K is 16 bytes in length and is made up as $K = K_1 \| K_2$ (the concatenation of the two eight bytes DES keys K_1 and K_2). Assuming that $K_1 = K_L$ and $K_2 = K_R$, each triple-DES key can also be expressed as $K = K_L \| K_R$.

b8...b1
b8...b1

Byte	16	15	14	13	12	11	10	9		8	7	6	5	4	3	2	1
	$K_L = K_1$									$K_R = K_2$							

Figure 5: structure of a triple-DES key

In the above described structure of the triple-DES, the following applies:

- Byte 1 is the Least Significant Byte of KR and also the Least Significant Byte of K
- Byte 8 is the Most Significant Byte of KR
- Byte 9 is the Least Significant Byte of KL
- Byte 16 is the Most Significant Byte of KL and also the Most Significant Byte of K
- Bit 1 is the Least Significant Bit of any byte constituting KL or KR
- Bit 8 is the Most Significant Bit of any byte constituting KL or KR

9.10 Generation of the Master Keys

The procedures and mechanisms and the equipment used by the OBU manufacturer to generate the master keys are not described in this specification. Nevertheless it is highly recommended to rely on a robust and proven hardware-based random number generator in order to generate distinct and independent master keys for each operator.

For each operator, 5 unique triple-DES master keys are generated for each of the two types of keys to be stored in the OBU, resulting in a total of 10 distinct triple-DES master keys for each operator.

Therefore for each generation of keys (G denotes the key generation number), two (2) master keys are needed:

- one (triple-DES) master key, referenced as MKEY-N/G, from which OBU specific keys to be used to compute Issuer MACs are derived;
- one (triple-DES) master key, referenced as MKEY-F/G, from which OBU specific keys to be used to compute EFC operator MACs are derived;

Type of key	KEY GENERATION NUMBER				
	1	2	3	4	5
Master Key used to derive the OBU specific key for Issuer MACs	MKEY-N/1	MKEY-N/2	MKEY-N/3	MKEY-N/4	MKEY-N/5
Master Key used to derive the OBU specific key for EFC operator MACs	MKEY-F/1	MKEY-F/2	MKEY-F/3	MKEY-F/4	MKEY-F/5

Table 4 References for Master Keys

9.11 Distribution of the Master Keys

9.11.1 Availability of the master keys in the RSEs and the CSs

In order to allow the RSE and the CS to check the integrity of the data transmitted by the OBUs, the master keys used by OBU manufacturer in the generation of the OBU specific keys need also to be stored in the respective RSEs and CSs. These keys need therefore to be transmitted (distributed) securely to the operators concerned, in compliance with the relevant security requirements defined in [1] (particularly SR-OBU-13).

Assuming that OBUs designated as OBU-N(i) are manufactured for a Operator N (the native operator) operating a system equipped with RSE-N and CS-N, and that interoperability is required with an Operator F (the foreign operator), operating a system equipped with RSE-F and CS-F, the following tables define the availability of the master keys in the different sub-systems. The sign √ indicates that the key is present in the sub-system.

KEY GENERATION NUMBER	
G (NOTE 1)	
MKEY-N/1	
MKEY-F/1	√

Table 10 Availability of the master keys in RSE-N

KEY GENERATION NUMBER									
1		2		3		4		5	
MKEY-N/1	√	MKEY-N/2	√	MKEY-N/3	√	MKEY-N/4	√	MKEY-N/5	√
MKEY-F/1	√	MKEY-F/2	√	MKEY-F/3	√	MKEY-F/4	√	MKEY-F/5	√

Table 51 Availability of the master keys in CS-N

NOTE: The availability of the master keys at the CS depends very much on whether the CS is responsible for distributing the master keys to the RSE, and whether the CS performs additional checks on the validity of the transaction already checked by the RSE.

RSE-F	
KEY GENERATION NUMBER (G)	
MKEY-F/G (NOTE 1)	√

CS-F	
KEY GENERATION NUMBER (G)	
MKEY-F/G (NOTE 2)	Optional (Note 3)

Table 62 Availability of the master keys in RSE-F and CS-F

Note 1: In compliance with the security requirements defined in [1] (SR-RSE-9), only one generation of keys is stored in the RSE at any given time, starting with the first generation (number 1).

Note 2: The availability of the master keys at CS-F depends very much on whether CS-F is responsible for distributing the master keys to RSE-F, and whether the CS-F performs additional checks.

Note 3: Interoperability does not require the availability of the master key in CS-F.

9.11.2 Distribution of the master keys and loading mechanisms

The procedures and mechanisms used to distribute and load the master keys in the RSEs and the CSs are not covered by this specification.

9.12 Generation of the cryptographic keys to be stored in the OBU

Each OBU stores 5 generations of keys, and each generation of keys ("G" denotes the key generation number and "i" denotes the current OBU) contains 2 triple-DES keys:

- one OBU specific triple-DES key to be used to compute native service provider MACs, referenced as OBUKEY-N/G(i)
- one OBU specific triple-DES key to be used to compute foreign service provider MACs,
  referenced as OBUKEY-F/G(i)

Type of key	KEY GENERATION NUMBER				
	1	2	3	4	5
OBU specific key for Issuer MACs	OBUKEY-N/1(i)	OBUKEY-N/2(i)	OBUKEY-N/3(i)	OBUKEY-N/4(i)	OBUKEY-N/5(i)
OBU specific key for EFC operator MACs	OBUKEY-F/1(i)	OBUKEY-F/2(i)	OBUKEY-F/3(i)	OBUKEY-F/4(i)	OBUKEY-F/5(i)

Table 13 References for the OBU specific keys

Each OBU specific key is computed as follows:

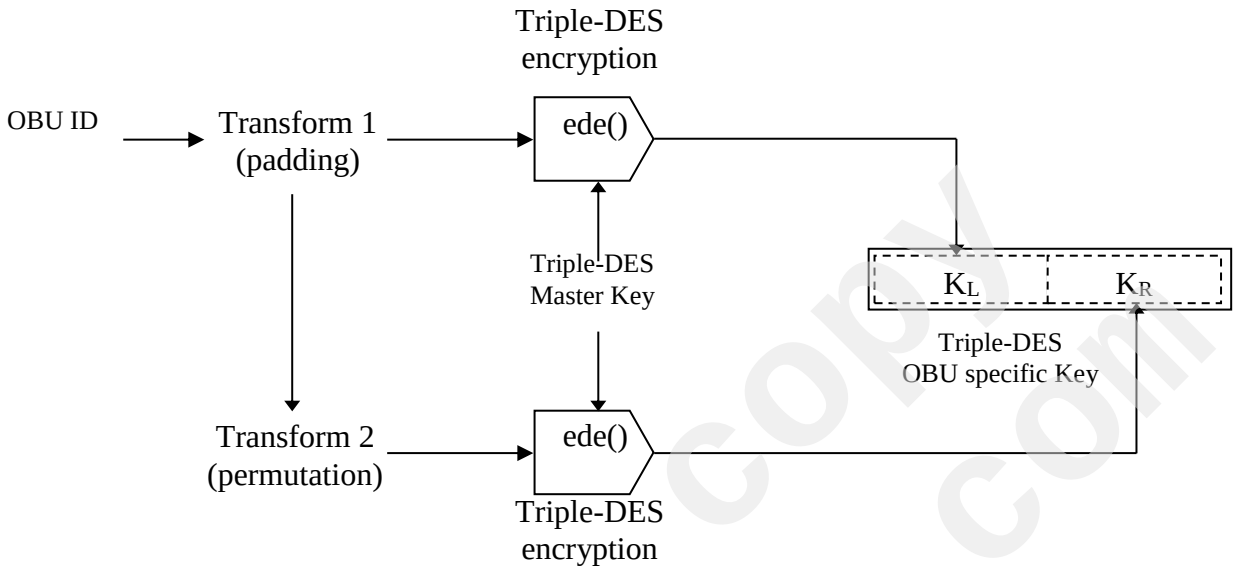


Figure 6: General scheme used to compute the OBU specific keys

9.12.1 Generation of the keys OBUKEY-N/G(i)

Use the following procedure to compute a key OBUKEY-N/G(i) for a given generation:

1) Get the OBU ID (which is encoded in 7 bytes) for the on-board unit OBU(i)
2) Pad left the OBU ID with the byte 'FF' to obtain an 8 bytes value VAL

   VAL = OBU ID || 'FF'

   Example:
   Assuming that OBU ID = '11 22 33 44 55 66 77' then VAL = 'FF 11 22 33 44 55 66 77'
3) Perform a permutation of all bytes of the value VAL to obtain the value PVAL:

   PVAL = '77 66 55 44 33 22 11 FF'
4) Compute the first part K_L of OBUKEY-N/G(i) as follows:

   $K_L = ede_{MKKEY\text{-}N/G}\ (VAL)$
5) Compute the second part K_R of OBUKEY-N/G(i) as follows:

   $K_R = ede_{MKKEY\text{-}N/G}\ (PVAL)$
6) Concatenate K_L and K_R to obtain OBUKEY-N/G(i):

   $OBUKEY\text{-}N/G(i) = K_L \,||\, K_R$
7) Repeat the same procedure for each generation of the key OBUKEY-N/G(i)

9.12.2 Generation of the keys OBUKEY-F/G(i)

Use the following procedure to compute a key OBUKEY-F/G(i) for a given generation:

1) Get the OBU ID (which is encoded in 7 bytes) for the on-board unit OBU(i)
2) Pad left the OBU ID with the byte 'FF' to obtain an 8 bytes value VAL

   VAL = OBU ID || 'FF'

   Example:
   assuming that OBU ID = '11 22 33 44 55 66 77' then VAL = 'FF 11 22 33 44 55 66 77'
3) Perform a permutation of all bytes of the value VAL to obtain the value PVAL:

   PVAL = '77 66 55 44 33 22 11 FF'
4) Compute the first part K_L of OBUKEY-F/G(i) as follows:

   $K_L = ede_{MKKEY\text{-}F/G}\ (VAL)$
5) Compute the second part K_R of OBUKEY-F/G(i) as follows:

   $K_R = ede_{MKKEY\text{-}F/G}\ (PVAL)$
6) Concatenate K_L and K_R to obtain OBUKEY-F/G(i):

   $OBUKEY\text{-}F/G(i) = K_L \,||\, K_R$
7) Repeat the same procedure for each generation of the key OBUKEY-F/G(i)

9.13 Computation of the Message Authentication Codes

Using the key generation number KGN and the random number RND-1 transmitted by the RSE, the OBU applies the following basic procedure to generate the Message Authentication Codes MAC1 and MAC2:

1) Compose the message M by concatenating the values OBU ID, OBU STATUS, TC and RND-2

   M = OBU ID || OBU STATUS || TC || RND-2

   Example:
   assuming that
   OBU ID = '11 22 33 44 55 66 77'
   OBU STATUS = '00 00'
   TC = '12 34'
   RND-2 = C2 F7 A3 E5'
   then
   M = '11 22 33 44 55 66 77 00 00 12 34 C2 F7 A3 E5'
2) Pad right the last block of M with as few 0-bits as necessary to obtain a multiple of eight bytes blocks.

   Example:
   assuming that
   M = '11 22 33 44 55 66 77 00 00 12 34 C2 F7 A3 E5'

then
M = '11 22 33 44 55 66 77 00 00 12 34 C2 F7 A3 E5 00'

3) Select the OBU specific key OBUKEY-N/G(i) corresponding to the key generation number KGN

4) Build up a 16 bytes time variant parameter TVP by concatenating RND-1 with the time parameter contained in the BST, and repeating the process

TVP = RND-1 || TIME || RND-1 || TIME

Where TIME is given the value sent by the RSE in the parameter Time of the BST

Example:
assuming that
RND-1 = '49 28 F3 C2'
Time = '01 23 45 67'
then
TVP = '49 28 F3 C2 01 23 45 67 49 28 F3 C2 01 23 45 67'

5) Compute the triple-DES session key SSKEY-N corresponding to OBUKEY-N/G(i)

SSKEY-N = OBUKEY-N/G(i) ⊕ *TVP*

SSKEY-N can also be expressed as SSKEY-N = SSKEY-N_L || SSKEY-N_R

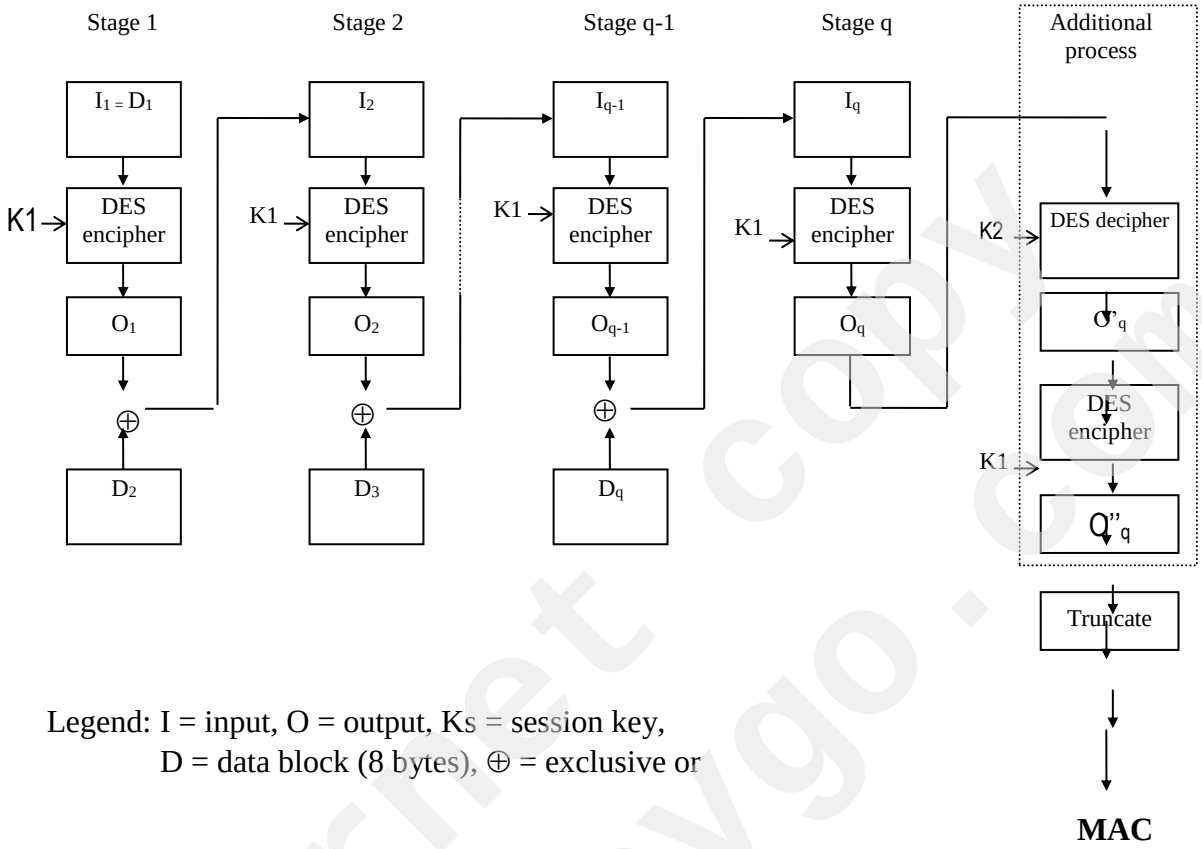
6) Compute the triple-DES session key SSKEY-F corresponding to OBUKEY-F/G(i)

SSKEY-F = OBUKEY-F/G(i) ⊕ *TVP*

SSKEY-F can also be expressed as SSKEY-F = SSKEY-F_L || SSKEY-F_R

7) Using the appropriate session key (SSKEY-N for MAC1 and SSKEY-F for MAC2), compute the corresponding Message Authentication Codes MAC1 and MAC2 for the message M obtained in step 3.

The MAC is calculated according to ISO /IEC 9797:1994 (4), including Annex A-1 - Optional process 1, as illustrated in the figure below. Note that MAC is obtained by taking the leftmost 32 bits of the final block O''$_q$



Legend: I = input, O = output, Ks = session key,
D = data block (8 bytes), ⊕ = exclusive or

Figure 7: Method used to compute the MACs

NOTE: Only two stages are required in the computation of MAC1 or MAC2

To compute MAC1, apply: K1 = *SSKEY-*N_L and K2 = *SSKEY-*N_R

To compute MAC2, apply: K1 = *SSKEY-*F_L and K2 = *SSKEY-*F_R

8) Generate a random number RND-2

9) Increment internally the transaction counter (TC) after the transmission of the application data, MAC1 and MAC2. If the transaction counter overflows, reset the transaction counter to the value '0000'

9.14 Security mechanisms for the RSE

Although the specifications do not prescribe any particular design as for the implementation level solutions used in the RSE to satisfy the general security requirements defined for the RSE, the use of Secure Application Modules (SAM) is recommended for the storage of the cryptographic keys in the RSE.

9.15 Initialisation of the transaction by the RSE

The RSE applies the following security related procedure to initialise the transaction with the OBU via the DSRC interface:

1) Generate a random number,
2) Select the proper key generation, and
3) Transmit these parameters to the OBU using the command GET_SECURE.request

9.16 Checking the Message Authentication Code MAC2

Assuming that the appropriate cryptographic key (MKEY-F) is available in the RSE, and that the response by the OBU to the GET_SECURE.request command contains the application data OBU ID, OBU status, TC, and RND-2, and MAC2, the RSE follows the following procedure to check the Message Authentication Code MAC2:

1) Compute the OBU specific key following the procedure described in chapter 9.1.2 Generation of the keys OBUKEY-F/G(i)
2) Compute a reference Message authentication code MAC2' using the procedure described in chapter 9.2 Computation of the Message Authentication Codes
3) Compare the value of the MAC2' with the value MAC2 transmitted by the OBU, if both are equal, the application data transmitted by the OBU is considered valid and flagged as such, if not, the transaction is flagged correspondingly and appropriate actions are initiated according to the security policy for the system
   - If MAC2 and MAC2' do match, consider the OBU to be valid, and give the operation status the value OPSTATUS = '00'
   - If MAC2 and MAC2' do not match, consider the OBU not to be valid, give the operation status the value OPSTATUS = '02', and initiate appropriate actions

9.17 Context mark

Included in the VST returned by the OBU is the data element EFC-ContextMark. EFC-ContextMark includes a data element called ContextVersion, which is used in these specifications as the version number of the security scheme supported by the OBU.

The current security scheme specified in this specification document is assigned the following value:

ContextVersion = $[0000001]_2$

9.18 EFC functions for OBU identification

The security protocol implemented by the OBU and the RSE in relation with the identification of the OBU relies on two EFC functions:

- GET_SECURE.request
  used to transmit a random number and the key generation number from the RSE to the OBU
- GET_SECURE.response
  used to retrieve the application data (OBU ID, OBU status, transaction counter a random number, and the Message Authentication Codes) from the OBU

10 APPENDIX D – PISTA

The following chapter is copied from relevant parts of "TS3204-02A BroBizz PISTA".

A similar description can be retained from Øresundsbro Konsortiet.

10.1 Introduction – Transponder data

The Transponder memory structure contains three elements according to the table below:

Element	Application ID AID	Element ID EID	Element Access
System	0 (No application)	0	Password
BroBizz PISTA Application Element	1 (EFC)	1	No protection
BroBizz Application Element	1 (EFC)	2	No protection

Table 74 Transponder data

The **System Element** contains data related to the transponder and is not connected to any specific application.

The **BroBizz PISTA Application Element** contains data that shall be used by the TCs in an interoperable environment based on the PISTA specification.

The **BroBizz Application Element** contains data that shall be used only by the BroBizz system.

The following sections will describe in detail all data in the elements and attributes.

Access in the tables below refers to the access conditions for each attribute when accessed at a transaction in a ETC lane.

10.2 System Element

The System Element with contained attributes is created at initialisation of the transponder at factory.

The Attributes marked with (*) are automatically sent from the transponder to the roadside in the connection phase.

Element ID (EID) = 0, Application ID (AID) = 0 (No application), Element Access = Password (AttrID=120)

Attribute/Definition	AttrID	Type	Value Range (Bold =Set at factory)	Length (Octets)	Access
ManufacturerID (*) Identifies the Manufacturer of the TRP. Is included in the VST.	1	**Choice 2** Octet String (Variable)	02 00 01$_{16}$ - 02 3F FF$_{16}$ **02 00 03$_{16}$** (3=Combitech)	**1+2**	**Read Only**
ManufacturingSerialNo Manufacturing Serial Number of the TRP	2	**Choice 2** Octet String (Variable)	04 00 00 00 00 – 04 FF FF FF FF$_{16}$ **04$_{16}$ + See 10.2.1**	**1+4**	**Read Only**
EquipmentClass (*) Manufacturer specific equipment class. Is included in the VST.	3	**Choice 2** Octet String (Variable)	02 00 00$_{16}$ - 02 7F FF$_{16}$ **02 50 00$_{16}$ for current version of TS3204/02A**	**1+2**	**Read Only**
ActivityTimer Number of milliseconds OBE has been active	7	**Choice 2** Octet String (Variable)	04 00 00 00 00$_{16}$ – 04 FF FF FF FF$_{16}$ **04 00 00 00 00$_{16}$**	**1+4**	**Read Only**
OBEStatus (*) OBE status according to GSS	10	**Choice 2** Octet String (Variable)	02 00 00$_{16}$ - 02 FF FF$_{16}$ **02 00 00$_{16}$**	**1+2**	**Read/Write**
BatteryInsertionDate Date of battery insertion/ replacement	16	**Choice 2** Octet String (Variable)	02 00 00$_{16}$ - 02 FF FF$_{16}$ **02$_{16}$ + Number of days since 1st of January 1970.**	**1+2**	**Read Only**
OBEGroupID	17	**Choice 2** Octet String (Variable)	02 00 00$_{16}$ - 02 FF FF$_{16}$ **02$_{16}$ + Randomly generated value for each OBE**	**1+2**	**Read Only**
ElementPassword	120	**Choice 2** Octet String (Variable)	04 00 00 00 00$_{16}$ – 04 FF FF FF FF$_{16}$ **04$_{16}$ + See 10.2.2**	**1+4**	**No Access**

Table 85 System elements

10.2.1 ManufacturingSerialNo

The syntax of the ManufacturingSerialNo is YYWWNNNNNN, where:

YY denotes production year

WW denotes production week of that year

NNNNNN is the serial number that is unique within each week.

The ManufacturingSerialNumber is coded as follows into the four octets:

YY = production year in 2 BCD coded digits (8 bits)

WW = production week in binary code (6 bits)

NNNNNN = serial number in binary code (18 bits)

10.2.2 Element Access Password

The System Element Access Password belonging to BroBizz is personalised in factory to a four-octet value. The Password (4 bytes) is sent as the Access Credentials in the request to the TRP when attributes in the System Element is to be accessed.

10.3 PISTA Application Elements

The attributes are created at customisation of the transponder and the data in these attributes is written at personalisation. The Attributes marked with (*) are automatically sent from the transponder to the roadside in the connection phase.

Element ID (EID) = 1, Application ID (AID) = 1 (EFC), Element Access = No protection

Attribute	Attr ID	Type	Value Range (Bold =Set at factory)	Length in octets	Access
EFC Context Mark (*)	**0**	**Choice 32**	**97 80 03 00 01 02$_{16}$**	**6**	**No access**
ContractProvider Identifies the organisation that issued the service rights given in the Contract. Numbers are assigned on a national basis.		Provider=	AA..ZZ & 0..16383 **97 80 03$_{16}$** Country Code /Issuer ID	3	
TypeOfContract ContractProvider-specific designation of the rules that apply to the Contract.		Octet string(2)	00 00 – FF FF$_{16}$ **00 01$_{16}$**	2	
ContextVersion Denotes the implementation version of the concerned contract within the context of the given ContractProvider, value assigned at the discretion of the ContractProvider.		Integer (0..127,..)	00 – FF$_{16}$ **02$_{16}$**	1	
ContractAuthenticator	**4**	**Choice 36** Octet String (Variable)	**See 10.3.1**	**1 + 4**	**Read Only**
VehicleClass Service provider specific information pertaining to the vehicle.	**17**	**Choice 49** INT1	**00**	**1**	**Read/ Write**

Vehicle Dimensions VehicleLength Overall Nominal maximum overall length of the vehicle, in dm, rounded to the next dm. VehicleHeight Overall Nominal overall unladen height, in dm, rounded to the next dm. VehicleWidth Overall Nominal overall width, in dm, rounded to the next dm.	**18**	**Choice 50** INT1 INT1 INT1	**00 00 00$_{16}$** **00$_{16}$** **00$_{16}$** **00$_{16}$**	**3** 1 1 1	**Read/ Write**
VehicleAxles	**19**	**Choice 51**	**00 00$_{16}$**	**2**	**Read/ Write**
Attribute	**Attr ID**	**Type**	**Value Range (Bold =Set at factory)**	**Length in octets**	**Access**
VehicleFirstAxle Height Bonnet height, measured over the front axle, in dm, rounded to next dm. VehicleAxles Number Number of axles, including drop axle.		INT1 Vehichle-AxlesType	**00$_{16}$** **00$_{16}$**	1 1	
VehicleAuthenticator Authenticator calculated on data elements when entered or modified. Shall be the result of a cryptographic calculation using all the vehicle attributes.	**23**	**Choice 55** Octet string (Variable)	**04 00 00 00 00$_{16}$** Length + Data	**5**	**Read/ Write**
EquipmentOBUId Identification number of onboard unit defined by its manufacturer.	**24**	**Choice 56** Octet string (Variable)	**04 NN NN NN NN$_{16}$** **Length+Manufacturing Serial Number, See 10.2.1**	**1+4**	**Read Only**
EquipmentStatus	**26**	**Choice 58** BIT STRING (16)	**00 00$_{16}$**	**2**	**Read/ Write**
PaymentMeans	**32**	**Choice 64**	**92 08 60 62 NN NN NN NL FF FF DB 9F 00 00$_{16}$**	**14**	**Read Only**

PersonalAccountNumber Coded according to financial institutions. (Imported from credit cards ISO 7812)		Personal Account-Number	**92 08 60 62 NN NN NN NL FF FF$_{16}$** **See 10.3.2**	10	
PaymentMeansExpiryDate		Date-Compact	**DB 9F$_{16}$** 31 Dec 2099	2	
PaymentMeansUsageControl		Octet string (2)	**00 00$_{16}$**	2	
ReceiptData1	**33**	**Choice 65**	**00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00$_{16}$**	**28**	**Read/ Write**
SessionTime Date and time of session with a two-seconds resolution.		DateAnd-Time	**00 00 00 00$_{16}$**	4	
SessionServiceProvider Organisation that provides the service of the session.		Provider	AA .. ZZ & 0 .. 16383 **00 00 00$_{16}$**	3	
Attribute	**Attr ID**	**Type**	**Value Range (Bold =Set at factory)**	**Length in octets**	**Access**
StationLocation Service provider specific coding of the station location.		INT2	0..9999 **00 00$_{16}$**	2	
SessionLocation Service provider specific coding of the session location within the station location		INT1	0/1 + 0..127 **00$_{16}$**	1	
TypeOfSession Designates the type of service station.		INT1	**00$_{16}$**	1	
SessionResult Code designating whether a session has been completed successfully or not.		ResultOp	**00$_{16}$**	1	
SessionTarifClass Service provider specific tariff class applied in the session.		INT1	**00$_{16}$**	1	

SessionClaimedClass Service provider specific vehicle class derived from claimed characteristics in the data group VehicleClass		INT1	**00_{16}**	1	
SessionFee		PaymentFee	**$00\ 00\ 00\ 00_{16}$**	4	
SessionContractProvider		Provider	**$00\ 00\ 00_{16}$**	3	
SessionTypeOfContract Contrac provider specific designation of the rules that apply to the Contract.		Octet string(2)	**$00\ 00_{16}$**	2	
SessionContextVersion It identifies the version of the transaction model used formatting the data.		INT1	**00_{16}**	1	
RecieptAuthenticator		Octet string(4)	**$00\ 00\ 00\ 00_{16}$**	4	
ReceiptData2	**34**	**Choice 66**	**$00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00_{16}$**	**28**	**Read/ Write**
SessionTime Date and time of session with a two-seconds resolution.		DateAnd-Time	**$00\ 00\ 00\ 00_{16}$**	4	
SessionServiceProvider Organisation that provides the service of the session.		Provider	AA .. ZZ & 0 .. 16383 **$00\ 00\ 00_{16}$**	3	
Attribute	**Attr ID**	**Type**	**Value Range (Bold =Set at factory)**	**Length in octets**	**Access**
StationLocation Service provider specific coding of the station location.		INT2	0..9999 **$00\ 00_{16}$**	2	
SessionLocation Service provider specific coding of the session location within the station location		INT1	0/1 + 0..127 **00_{16}**	1	
TypeOfSession Designates the type of service station.		INT1	**00_{16}**	1	

SessionResult Code designating whether a session has been completed successfully or not.		ResultOp	**00_{16}**	1	
SessionTariffClass Service provider specific tariff class applied in the session.		INT1	**00_{16}**	1	
SessionClaimedClass Service provider specific vehicle class derived from claimed characteristics in the data group VehicleClass		INT1	**00_{16}**	1	
SessionFee		PaymentFee	**$00\ 00\ 00\ 00_{16}$**	4	
SessionContractProvider		Provider	**$00\ 00\ 00_{16}$**	3	
SessionTypeOfContract Contrac provider specific designation of the rules that apply to the Contract.		Octet string(2)	**$00\ 00_{16}$**	2	
SessionContextVersion It identifies the version of the transaction model used formatting the data.		INT1	**00_{16}**	1	
RecieptAuthenticator		Octet string(4)	**$00\ 00\ 00\ 00_{16}$**	4	
Private 1 (SB, LPN)	**91**	**Octet String (Variable)**	**$0A\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00_{16}$**	**1+10**	**Read/ Write**
Private 2 (SB, Shadow class)	**92**	**Octet String (Variable)**	**$01\ 00_{16}$**	**1+1**	**Read/ Write**
Private 3 (SB, Reserv)	**93**	**Octet String (Variable)**	**$06\ 00\ 00\ 00\ 00\ 00\ 00_{16}$**	**1+6**	**Read/ Write**
Attribute **Private 4 (SB, Black List)**	**Attr ID** **94**	**Type** **Octet String (Variable)**	**Value Range (Bold =Set at factory)** **$01\ 00_{16}$**	**Length in octets** **1+1**	**Access** **Read/ Write**
Application Element Authentication Key 1	**111**	**Octet String (Variable)**	**See 10.3.3**	**1+16(1)**	**No access**

Application Element Authentication Key 2	**112**	**Octet String (Variable)**	**See 10.3.3**	**1+16[(1)]**	**No access**
Application Element Authentication Key 3	**113**	**Octet String (Variable)**	**See 10.3.3**	**1+16[(1)]**	**No access**
Application Element Authentication Key 4	**114**	**Octet String (Variable)**	**See 10.3.3**	**1+16[(1)]**	**No access**
Application Element Authentication Key 5	**115**	**Octet String (Variable)**	**See 10.3.3**	**1+16[(1)]**	**No access**
Application Element Authentication Key 6	**116**	**Octet String (Variable)**	**See 10.3.3**	**1+16[(1)]**	**No access**
Application Element Authentication Key 7	**117**	**Octet String (Variable)**	**See 10.3.3**	**1+16[(1)]**	**No access**
Application Element Authentication Key 8	**118**	**Octet String (Variable)**	**See 10.3.3**	**1+16[(1)]**	**No access**

Table 96 PISTA application elements

[(1)] The TRP always perform Authentication using 3-DES but since the two 8 octet halves in the 16 octets Application Element Keys are equal this will result in DES.

10.3.1 ContractAuthenticator

The first octet in this Attribute is a lenghtbyte. Therefore the value of the first octets is 0x04.

The next four octets is calculated according to the following procedure:

1. Let the first 4 octets of EFC Attribute EFCContextMark be CA_1.
2. Let the last 4 octets of customer defined PAN (including Luhn) be CA_2.
3. Get the attribute ContractAuthenticator with the following algorithm:
   ContractAuthenticator = [CA_1] XOR [CA_2]

10.3.2 Personal Account Number (PAN)

The PAN number value is defined in ISO 7812-1 and it is coded as BCD characters into an octet string with length 10 as follows:

II II II NN NN NN NN NL FF FF_{16} where:

IIIIII is the 6-digit Issuer Identification Number.

NNNNNNNNN is a 9-digit Individual Account Identification Number:

For BroBizz the value shall be: 92 08 60 62 NN NN NN NL where NN NN NN N is provided by BroBizz.

L is a Luhn key calculated over the PAN number. FF FF are fill characters set to the value **FF FF$_{16}$**

10.3.3 Authentication Keys

The Authentication Keys in the EFC element shall be derived from the Master Keys that belong to BroBizz. Use the following procedure to compute the AuthenticationKey(i) for a given generation (i):

1. Let the first 8 octets of the EFC Attribute PaymentMeans be PM_8.
2. Get the attribute Compact_PaymentMeans by truncating the 64 bits PM_8 to 32 bits with the following algorithm:

Compact_PaymentMeans = [HighDWord32(PM_8)] XOR [LowDWord32(PM_8)]

3. Get the first 3 octets of EFC-ContextMark, i.e. the ContractProvider
4. Pad left the concatenation of Compact_PaymentMeans || ContractProvider with '00' to obtain an 8 bytes value VAL:

VAL = 'Compact_PaymentMeans || ContractProvider || 00'

5. Compute the AuthenticationKey(i) as follows: AuthenticationKey(i) = TDES $_{\text{MAuthenticationKey(i)}}$(VAL)
6. The Application Element Authentication Key_(i) to be programmed into the transponder is then created by concatenating the AuthenticationKey(i) to itself to form a 16-byte long word with two equal halves.

10.4 BroBizz Application Element

The attributes are created at customisation of the transponder and the data in these attributes is written at personalisation. The Attributes marked with (*) are automatically sent from the transponder to the roadside in the connection phase.

Element ID (EID) = 2, Application ID (AID) = 1(EFC), Element Access = No protection

Attribute	Attr ID	Type	Value Range (Bold =Set at factory)	Length in octets	Access
EFC Context Mark (*)	**0**	**Choice 32**	**97 80 03 00 01 01$_{16}$**	**6**	**No access**
ContractProvider.		Provider=	AA..ZZ & 0..16383 **97 80 03$_{16}$** Country Code = Issuer ID =	3	
TypeOfContract .		Octet string(2)	00 00 – FF FF$_{16}$ **00 01$_{16}$**	2	

ContextVersion		Integer (0..127,..)	00 – FF_{16} **01_{16}**	1	
TRP ID	**101**	**Octet String (Variable)**	08 00 00 00 00 00 00 00 00– 08 FF FF FF FF FF FF FF FF$_{16}$ **Same as PAN in EID1** **See 10.3.2**	**1+8**	**Read Only**
Spare Attribute	**102**	**Octet String (Variable)**	**0A 00 00 00 00 00 00 00 00 00 00$_{16}$**	**1+10**	**Read/Write**
Application Element Authentication Key 1	**112**	**Octet String (Variable)**	**See 10.4.1**	**1+16**	**No access**
Application Element Authentication Key 2	**113**	**Octet String (Variable)**	**See 10.4.1**	**1+16**	**No access**
Application Element Authentication Key 3	**114**	**Octet String (Variable)**	**See 10.4.1**	**1+16**	**No access**
Application Element Authentication Key 4	**115**	**Octet String (Variable)**	**See 10.4.1**	**1+16**	**No access**

Table 107 BroBizz application elements

10.4.1 Element Authentication Keys

The Authentication Keys in the EFC element shall be derived from the Master Keys that belong to BroBizz. These keys shall be derived in the following way:

Authentication Key = 3-DES(Master Authentication Key, TRP ID_{BCD}).

11 APPENDIX E – BroBizz

The following chapter is copied from relevant parts of "TRP-TS3204 BroBizz".

11.1 Overview

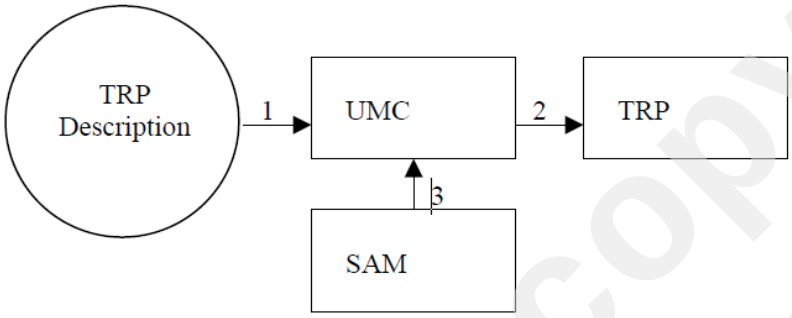


Figure 8 External interface diagram for transponder description.

No.	Interface Name	Brief interface description
1.	SOB file	SOB file describes the contents of user memory in the transponder, it's interpreted by UMC that generates the personalisation. See *[Ref 8]*
2.	TRP Image	UMC generates the personalisation with a static image.
3	DLL TRSAM	UMC gets secret information from SAM card by calling functions in the DLL. See *[Ref 4]*.

11.2 Data and Data Structure

11.2.1 Main Domains

Domain	Requirements Description
SAM Data	Requirements on the data for the SAM-System
Issuer Key	Requirements on the data for the key used for generation of AccessCredentials for access to the TRP
System Element	Requirements on the data for the TRP System Element
BroBizz Application Element	Requirements on the data for the BroBizz AFC Element

Table 118 Data structure -main domains

11.2.2 SAM Context Mark

R1 SAM Context Mark is 6-bytes of § 4.2.6.1 EFC Context Mark.

11.2.3 SAM Password

R1 SAM Password is 8-bytes in EF_CHV, See CT-KEY-98:003 for the value.

11.2.4 Issuer key

R1 The value of the Issuer Key shall be retrieved from the SAM, using following command: Command = SAM_ISSUERKEY

Input Data = 3030_{16} (2-bytes File ID of EF_Issuer_Key file)

11.2.5 System Element

System Element EID = 0, AID=0, Password (attribute 120)

Attribute	AttID	Data Size	Type	Value	Access
ManufactureID	1	2	Octet String	00 03	No Access
ManufactureSerialNo	2	4	Octet String	See *[Ref 6]*	Read only
EquipmentClass	3	2	Octet String	50 00_{16}	No Access
ActivityTimer	7	4	Octet String	00 00 00 00_{16}	Read only
OBEStatus	10	2	Octet String	00 00_{16}	Write only
BatteryInsertionDate	16	2	Octet String	See 112.5.1	Read only
ElementPassword	120	4	Octet String	See 114.2.5.2	No Access

Table 129 Data structure -system elements

11.2.5.1 Battery Insertion Date

R1 The value of Battery Insertion Date is a 2-byte binary coded integer, denoting the number of days relative to 1970-01-01 when the battery was inserted.

Command = BAT

11.2.5.2 Element Password

R1 The value of the Element Password is retrieved from the SAM, using following command: Command = SAM_READ_FILE

Input Data = 4040_{16} (2-bytes File ID of EF_Element_Password)

11.2.6 BroBizz Application Element

Common EFC Element EID = 1, AID=1, No Protection

Attribute	AttID	Data Size	Type/Choice	Value	Access	Note
EFC Context Mark	0	6	Octet String	See 11.2.6.1.	No Access	
TRP ID	101	4	Octet String	See 11.2.6.2	Read	
TRP Authentication Key 1	112	16	Octet String	See 11.2.6.3	No Access	
TRP Authentication Key 2	113	16	Octet String	See 11.2.6.3	No Access	
TRP Authentication Key 3	114	16	Octet String	See 11.2.6.3	No Access	
TRP Authentication Key 4	115	16	Octet String	See 11.2.6.3	No Access	
Spare Attribute 1	116	30	Octet String	Zero	Read/ Write	
Spare Attribute 2	117	30	Octet String	Zero	Read/ Write	
Spare Attribute 3	118	30	Octet String	Zero	Read/ Write	
Spare Attribute 4	119	30	Octet String	Zero	Read/ Write	
Spare Attribute 5	120	10	Octet String	Zero	Read/ Write	

Table 20 BroBizz Application Element

11.2.6.1 EFC Context Mark

R1 The EFC Context Mark does consist of

- Country Code = ITA2-code for DK
- Issuer ID = 3
- Type of Contract = 0000_{16}
- Context Version = 01_{16}

The total EFC-Context Mark is: 97 80 03 00 00 01_{16}

11.2.6.2 TRP ID

R1 Ref. *[Ref 9]*, *[Ref 10]* derived

The value of the TRP ID shall have the following format

For the BroBizz application the number of BCD-coded digits of the TRP ID shall be fixed to 9. This means that the length of the TRP ID will be 8 bytes.

The first 3 bytes correspond to the first 3 bytes of the AFC Context Mark.

Issuer						Individual Account Identification									Check digit
9	7	8	0	0	3	BCD	BCD	BCD	BCD	BCD	BCD	BCD	BCD	BCD	BCD

Table 21 BroBizz TRP ID

R2 Ref. *[Ref 9]*, *[Ref 11]* derived

The check digit shall be computed using the Luhn Modulus 10 formula.

R3 Ref. *[Ref 10]*

The first digit in the Individual Account Identification shall denote the Vehicle Class: 0=passenger car, 1=lorry.

11.2.6.3 Element Authentication Key 1-4

R1 The value of Element Authentication Key 1-4 is obtained with the following command Command =SAM_DIV_KEY

Input Data = TRP ID (See 11.2.6.2)

key number = 00_{16} for key 1
key number = 02_{16} for key 2
key number = 04_{16} for key 3
key number = 07_{16} for key 4

11.3 Referenced documents in this chapter

No.	Identification	Name or Description
[Ref 1]	8633 800-024	SRS TRP TS3204
[Ref 2]	AL-00:004	Project Terminology Description (PTD) for AL
[Ref 3]	SCT-QAR-4.4.1-002	Programvaruutveckling
[Ref 4]	8633 800-338	IRS DLL for TRSAM
[Ref 5]	8633 900-187	PPI for TRP-Production SAM, BroBizz
[Ref 6]	8633 900-134	Labels and Serial Number for TS3204/00A, SBP
[Ref 7]	CTD-00:001	Company Terminology Description (CTD)
[Ref 8]	8633 800-702	IRS for UMC
[Ref 9]	ISO/IEC 7812-1	Identification cards-Identification of Issuers - Part 1: Numbering system
[Ref 10]	CT-SBP-98:074	Transponder ID numbering – BroBizz

[Ref 11]	TS3200-149	LUHN Formula (Mod 10) for Validation of a Primary Account Number
[Ref 12]	8633 800-025	SDD TRP TS3204
[Ref 13]	8633 900-341	Manufacturing Serial Number for TS3204
[Ref 14]	CT-KEY-98:003	CHV 1 of SAM 8633 000-493, BroBizz

12 APPENDIX F - References

12.1 Standards and external documents

For dated references, subsequent amendments to or revisions of any of these publications apply only when incorporated in it by amendment or revision. For undated references the latest edition of the publication referred to applies.

Reference	Document Ref	Date / Version	Document title
[IAP]	EN 15509	2014	Road Traffic and Transport Telematics (RTTT) – Electronic Fee Collection – Interoperability application profile for DSRC
[EFC API]	EN ISO 14906:2011/ Amd1:2015	2011/ Amd1:2015	Road Traffic and Transport Telematics (RTTT) –Electronic Fee Collection – Application interface definition for dedicated short range communication
[GSS]	GSS	V3.2:2003	Global Specification for Short Range Communication (Kapsch TrafficCom AB, Kapsch Telecom GmbH, Thales e-Transactions CGA SA, version 3.2, 2003-08, http://www.etc-interop.com/pdf/gss_32.pdf)
[L1]	EN 12253		Road Transport and Traffic Telematics (RTTT) –Dedicated Short-Range Communication (DSRC) –Physical layer using microwave at 5.8 GHz
[L2]	EN 12795		Road Transport and Traffic Telematics (RTTT) – Dedicated Short-Range Communication (DSRC) – DSRC data link layer: Medium access and logical link control
[L7]	ISO 15628 (formerly EN 12834)	2013	Road Transport and Traffic Telematics (RTTT) – Dedicated Short-Range Communication (DSRC) – DSRC Application Layer (formerly EN 12834)
[Profiles]	EN 13372		Road Transport and Traffic Telematics (RTTT) – Dedicated Short-Range Communication (DSRC) – Profiles for RTTT applications

Reference	Document Ref	Date / Version	Document title
[UNECE]			ECONOMIC COMMISSION FOR EUROPE - INLAND TRANSPORT COMMITTEE - Working Party on the Construction of Vehicles TRANS/WP.29/78/Rev.1/Amend.2 - CONSOLIDATED RESOLUTION ON THE CONSTRUCTION OF VEHICLES (R.E.3)
[Reg_doc]			Directive 1999/37/EC on Registration Documents

Table 22 Standards and external documents

12.2 EasyGo Documents

Reference	Document Ref	Date / Version	Document title
[EasyGo-202]			EasyGo Roadside and On Board Equipment
[EasyGo-202-A]			EasyGo+ and EETS OBE Functional Requirements (this document) (Replacements for "Functional requirements for EasyGo+ OBEs")
[EasyGo-202-B]			EasyGo+ and EETS DSRC Tolling Data Specification (Replacement for "EasyGo+ OBE personalisation, configuration and operating parameters")
[EasyGo-202-C]			EasyGo+ and EETS DSRC transaction for Tolling and Enforcement (Replacement for "EasyGo+ DSRC transaction for tolling and enforcement")
[EasyGo-202-E]			EasyGo+ and EETS Acceptance Procedures (Replacements for "EasyGo+ OBE compatibility tests") To be replaced by revised document 207
[EasyGo-207]			EasyGo test strategy

Table 23 EasyGo documents