

# Basic Requirements Specification for Interoperable EFC-DSRC Systems in Sweden

## A Specification for Implementation of PISTA and CARDME

Document nature: General Technical Specification (Allmän Teknisk Beskrivning, ATB)

Document version: 1.0

Status: Approved

Date of issue: 12 November 2003

Author(s):	Jesper ENGDAHL	Rapp Trans AG
	Johan HEDIN	Hybris Konsult AB
	Jonas SUNDBERG	SWECO VBB AB

Reviewed by: Christer Rydmell      SNRA

**Titel:** Basic Requirements Specification for Interoperable EFC-DSRC Systems in Sweden. A Specification for Implementation of PISTA and CARDME.

**Författare:** Jesper Engdahl Rapp Trans AG, Johan Hedin Hybris Konsult AB, Jonas Sundberg Sweco VBB AB.

**Dokumentbeteckning:** Publikation 2003:155

**Utgivningsdatum:** 2003-11

**ISSN:** 1401-9612

**Distributör:** Vägverket, Butiken, 781 87 Borlänge. Telefon 0243-755 00, Fax 0243-755 00,

**e-post:** [vagverket.butiken@vv.se](mailto:vagverket.butiken@vv.se)

## Förord

ATB EFC är en allmän teknisk beskrivning (ATB) som innehåller Vägverkets krav på upphandling och utformning av elektroniska system för bilavgifter<sup>1</sup>. Syftet är att garantera teknisk interoperabilitet mellan bilavgiftssystem i Sverige.

ATB EFC omfattar krav på fordonsenheter (OBU), vägsidesutrustning (RSE) och kommunikation dem emellan (EFC transaktionen) vilka är formulerade i föreliggande dokument. Då upphandlingar av EFC-system oftast genomförs på engelska är hela denna ATB författad på engelska.

ATB EFC skall användas vid upphandlingar i Sverige (där Vägverket medverkar) av bilavgiftssystem för uppbörd av trängselavgift eller för betalning av vägtull, vilka påbörjas fr.o.m. den 2003-12-01. ATB EFC avser i nuläget inte eventuella uppbördssystem för skatt på tunga fordon på det allmänna vägnätet, i de fall de baseras på annan teknik än DSRC.

ATB EFC kan vid behov revideras, men skall i sådana fall beakta att tidigare upphandlade EFC-system, enligt denna ATB, är kompatibla med senare versioner.

I inledningen av dokumentet förklaras mer ingående dess avgränsning och i vilket sammanhang denna ATB tillämpas.

Borlänge i november 2003

Ingemar Skogö

---

<sup>1</sup> Förkortas här med det engelska 'EFC'. Se kapitel 3 för förklaring av förkortningar och termer.

## To the user of this specification

This specification defines the basic requirements for interoperability between electronic fee collection (EFC) systems based on dedicated short-range communication (DSRC) in Sweden. The specification enables an interoperable EFC payment service based on central account and post-payment. The specification defines technical requirements for the on-board unit (OBU), roadside equipment (RSE) and the communication between them.

The specification does neither provide a complete set of system's requirements nor a full set of requirements for all aspects of interoperability, but focuses on necessary requirements for technical interoperability. For a more comprehensive definition of the context for the use of this specification, see chapter 1 (Overview).

As the use of EFC-DSRC is considered well known; motivation, examples or user guidance are not included in the main body of the specification, although some explanations and examples can be found in annexes.

For full understanding of the specification, a certain level of background knowledge in EFC and DSRC is required.

This specification is made publicly available by SNRA with the aim to establish the technical framework for interoperability between EFC systems in Sweden, and to foster interoperability with EFC systems outside Sweden.

The specification shall always be referred to whenever used. The Swedish EFC-DSRC specification may be issued in later versions. Hence, the version number shall always be referred to whenever used.

“Basic Requirements Specification for Interoperable EFC-DSRC Systems in Sweden” is the property of the Swedish National Road Administration (SNRA). It forms part of the ATB-series - general technical specifications issued by SNRA – regulating procurement and design of EFC-systems in Sweden.

Any questions about the specification should be addressed to:

**Swedish National Road Administration**

Attn. Mr. Christer Rydmell

S-781 87 Borlänge

Sweden

Tel. +46 243-750 00

Fax: +46 243-758 25

E-mail: [christer.rydmell@vv.se](mailto:christer.rydmell@vv.se)

## Table of Contents

<b>1</b>	<b>OVERVIEW .....</b>	<b>6</b>
1.1	Introduction .....	6
1.2	Purpose and use of this specification .....	6
1.3	Background and context .....	6
1.4	Main features.....	8
1.5	Definition of the EFC service.....	8
1.6	Scope.....	9
1.7	Structure and contents of the document.....	10
<b>2</b>	<b>REFERENCES .....</b>	<b>11</b>
<b>3</b>	<b>ABBREVIATIONS .....</b>	<b>13</b>
<b>4</b>	<b>DSRC REQUIREMENTS .....</b>	<b>14</b>
<b>5</b>	<b>EFC TRANSACTION REQUIREMENTS .....</b>	<b>15</b>
5.1	EFC transactions .....	15
5.1.1	PISTA transaction .....	16
5.1.2	CARDME transaction .....	17
5.2	DSRC L7 services and EFC functions.....	18
<b>6</b>	<b>SECURITY FEATURES.....</b>	<b>19</b>
6.1	Security features overview.....	19
6.2	Computation of authenticator.....	20
6.2.1	ContractAuthenticator authenticator.....	20
6.2.2	PaymentMeans authenticator .....	21
6.3	Access credentials.....	22
6.3.1	OBU computation of Access credentials .....	23
6.3.2	RSE computation of Access credentials.....	23
<b>7</b>	<b>DATA ATTRIBUTE REQUIREMENTS.....</b>	<b>24</b>
7.1	Overview .....	24
7.2	Personalisation requirements.....	25
<b>ANNEX A : EFC ARCHITECTURE.....</b>		<b>26</b>
A.1	Actors' perspective .....	26
A.2	Functional perspective.....	27
A.3	Physical perspective.....	30
<b>ANNEX B – EFC DSRC TRANSACTIONS .....</b>		<b>32</b>
B.1	Migration path considerations .....	32
B.2	Comparison of PISTA and CARDME .....	33
<b>ANNEX C - CLASSIFICATION CONCEPTS.....</b>		<b>36</b>
<b>ANNEX D - DATA SPECIFICATION.....</b>		<b>38</b>
<b>ANNEX E - SECURITY IMPLEMENTATION EXAMPLES .....</b>		<b>45</b>
E.1	Key derivations.....	45
E.2	Computation of authenticators .....	47
E.3	Computation of Access Credentials .....	49

## 1 Overview

### 1.1 Introduction

This document is a specification that states the basic requirements for interoperability between electronic fee collection (EFC) systems based on dedicated short-range communication (DSRC) in Sweden. This document is a General Technical Specification ("Allmän Teknisk Beskrivning", ATB) and thus forms part of the general regulations governed by SNRA concerning procurement of DSRC based EFC systems in Sweden (see foreword in Swedish).

The specification is designed as a set of basic requirements for on-board unit (OBU) and roadside equipment (RSE), for technical interoperability between systems. This means that:

- The specification only specifies the critical requirements for technical interoperability. This is not a full specification for the entire system. Further requirements need to be made for the non-critical interoperability parts and interfaces of the system.
- Local additions (e.g. data elements, solutions, requirements) can freely be added to this basic specification, as long as such local features do not conflict with the interoperable features, thus enabling a large degree of flexibility in the system design.
- For interoperability, procedural and contractual issues need to be defined in addition to, and based on, this specification (see also 1.6).

### 1.2 Purpose and use of this specification

The purpose of this specification is to define the critical elements for technical interoperability between DSRC based EFC systems in Sweden. This is achieved through specification of the basic requirements for system design.

This specification is to be used as a reference document when procuring EFC systems. Following this specification will ensure that relevant standards are used in systems using this specification.

The specification may also be used for defining the technical parts of a national (or international) memorandum of understanding (MoU) for interoperability between operators if such a MoU was to be agreed in Sweden (or with other operators in Europe).

### 1.3 Background and context

In 2002 the need for an interoperable EFC solution in Sweden resurfaced with the increased interest in implementation of DSRC based EFC systems. Several projects are currently in progress that may use this specification:

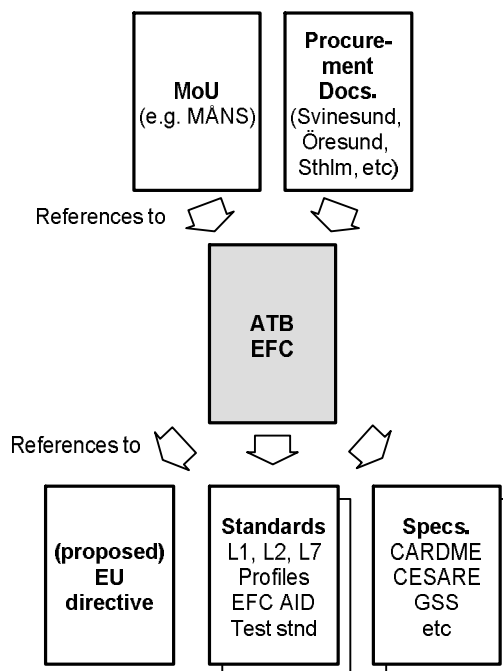
- **Öresund Bridge** (operates the BroBizz EFC system in co-operation with Store Bælt Bridge). Öresund Bridge is currently updating its EFC system for European interoperability based on the PISTA-specification.
- **Svinesund Bridge** will start operations of a fee collection system by 2005 (in co-operation with Norway). The EFC-system at Svinesund Bridge will be based on the PISTA-specification, and will in addition be able to handle also Norwegian AutoPass-clients.
- **Stockholm congestion charging** scheme aims to start operations in 2004/2005 for the Stockholm inner city region.

There are also discussions on the introduction of motorway fees on the E6-link in western Sweden as well as heavy goods vehicle (HGV)-fee on the road network in Sweden, at the time of writing of this specification.

During the last 10-15 years several projects and initiatives in Europe have worked with EFC interoperability. The results of these projects constitute a platform for this specification:

- Proposed **EU directive** on EFC. This defines the basic context for a European-wide EFC service.
- **MÅNS**. Nordic project that defines a MoU and a framework for (Nordic) interoperable EFC.
- **CESARE / PISTA**. ASECAP-led projects defining a full solution for interoperable EFC between ASECAP-operators.
- **CARDME**. EC-project defining an EFC transaction and a common EFC service.
- **CEN/ISO**. European and International standardisation bodies developing relevant standards.

The context for use of the specification is illustrated in the following figure:



**Figure 1.1** The context for the use of this specification.

There are three sets of documents that are referred to in this specification:

- [Draft EC Directive] is a general framework for EFC in Europe.
- Relevant standards, including DSRC standards ([EN L1], [EN L2], [EN L7], [EN Profiles], [ETSI DSRC] and [EFC AID]), test standards and standards concerning vehicles and electronics.
- References to other open specifications such as [CARDME], [CESARE], [PISTA] and [GSS].

Generally this specification refers to other documents, whenever applicable, rather than copy or rewrite similar solutions. This ensures a flexible approach to upgrades, effective use of achieved consensus results, and a concise specification.

The specification can be used as a reference document and as a basic specification in two main situations:

- When specifying and procuring a local EFC system in Sweden (or elsewhere), this specification defines the interoperable technical parts of the requirements specification.
- When making a MoU for interoperability (such as MÅNS for the Nordic countries), this specification may be used to define the interoperable technical parts of such a MoU.

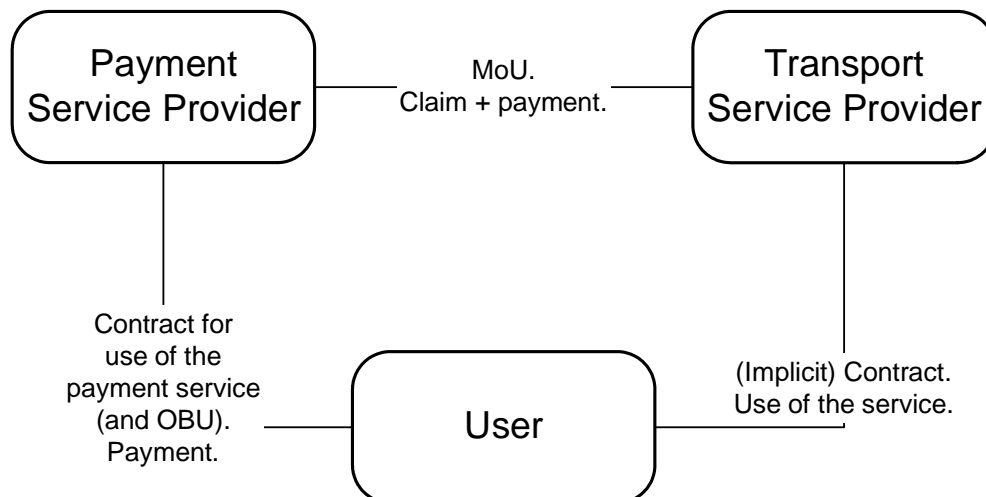
## 1.4 Main features

The main features of the specification are:

- Compliant with the proposed EU-directive on EFC.
- Compliant with CEN / ISO standards. Implementation of DSRC set B.
- Central account based on CESARE / PISTA and CARDME EFC transactions. Thus, this specification includes the PISTA-solution chosen for interoperability between Öresund, Stora Bält and Svinesund bridges.
- Full CARDME security scheme enabled.
- Interoperability between systems for motorway-fees and urban congestion charging.
- In line with EFC systems implemented in Sweden, Denmark, Austria, France and Spain that deploy mature industrial products from several suppliers.
- Manufacturer independent – all major vendors of OBU and RSE compliant with European DSRC 5.8 GHz can supply equipment according to this specification.
- Additional local solutions and transactions possible.
- Flexible classification scheme allowing for different solutions for users and operators.
- Clearly defined migration steps enabling operators (within a MoU) to implement defined additional security features.

## 1.5 Definition of the EFC service

The basic definition and structure of the interoperable EFC service is illustrated in the following figure. This figure describes the organisational framework, roles and entities in interoperable EFC:



**Figure 1.2** The basic definition and structure of the interoperable EFC service.

The basic roles (actors) in interoperable payment are:

- The **User** uses the transport service and pays for it by means of a payment service offered. The User is typically a vehicle owner.
- The **Payment Service Provider** (PSP, often called Issuer) is the entity responsible for the payment means (OBU, central account, service rights).
- The **Transport Service Provider** (TSP, often called EFC Operator) is the entity offering a transport service to the User (e.g. toll road access).

Please note that in many cases (e.g. in the Nordic countries today) the Transport Service Provider is also a Payment Service Provider. However, here when defining the generic roles in interoperable payment, this does not necessarily need be the case. A PSP may only offer the payment means and not being a TSP (e.g. a bank or a credit card company).

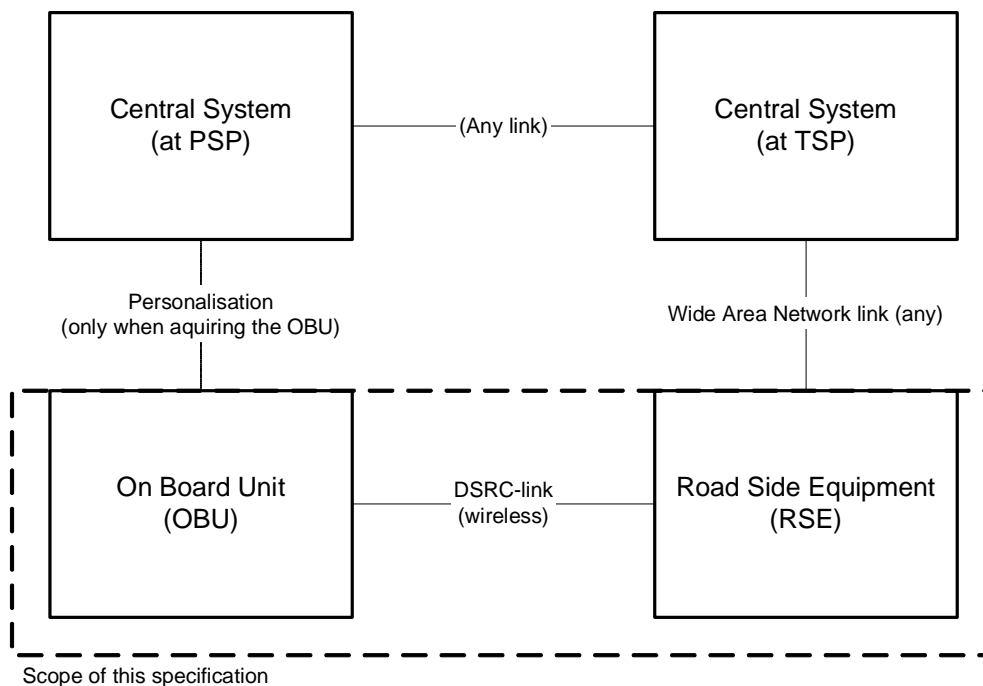
The main procedures in the interoperable EFC service are summarised:

- The User acquires an OBU and obtains a contract for use of the payment method from the Payment Service Provider.
- The User uses the road service and is charged a fee by the Transport Service Provider (the TSP decides on classification, possible enforcement, which contract to use, etc.).
- The TSP sends a claim for the amount to the PSP.
- The Users central account is settled by the Payment Service Provider (i.e. the User pays the fee) and the PSP pays the amount to the TSP.

A more comprehensive definition of the EFC service and its architecture is given in Annex A.

## 1.6 Scope

This specification deals only with parts of a full system design (or a full MoU) for interoperability. The specifications scope is mainly defined by the physical architecture as the specification focuses on the OBU, roadside equipment (RSE) and the interface between OBU / RSE. The specification has different requirements on OBU and RSE in many cases.



**Figure 1.3** Illustration of the scope of this specification.

Summary of scope (the parts that are within the specification):

- Payment method: Central account based on EFC-DSRC.
- The specification is primarily for use in Sweden, but with a clear aim to enable interoperability with other EFC systems in Europe.
- Physical systems: OBU, RSE and the interface between them.
- All functions and information flows related to the physical parts as above.

- All actors and responsibilities related to the physical parts as above.
- DSRC-link (for the interface as above).
- EFC transaction (for the interface as above).
- Data elements to be used by OBU and RSE.
- Security mechanisms for OBU and RSE.

The following aspects of EFC do not form part of this specification (but may benefit from the specification):

- Full requirements for procurement of an entire system.
- Contractual interoperability requirements (including MoU issues).
- Procedural interoperability requirements and organisation.
- Conformance procedures and test specification.
- Setting-up of central organisation (clearing operator, trusted third party (TTP), conformance test house etc.) etc.
- Legal issues.
- Other payment methods.
- Other basic technologies (e.g. global navigation and satellite system / cellular network or video registration).
- Other interfaces or functions in EFC than those specified above.
- Handling of and migration from local existing EFC systems (e.g. the current BroBizz).

Finally, SNRA plans to issue a Test Specification to be used for assessment of compliance with this specification.

## 1.7 Structure and contents of the document

This specification is deliberately written to be concise. Motivation, tutorial examples or user guidance are not included in the main body of the specification, but can partly be found in the annexes. For full understanding of this specification, a certain level of background knowledge in EFC and DSRC is required. Readers not familiar with the underlying specifications can acquire the necessary knowledge through consultation of the referenced documents. Laymen are advised to read tutorial documents, such as [CARDME, Parts 1-2], prior to reading this specification.

Structure and contents (Chapters 4-7 constitute the main body of the specification):

Chapter 1	Overview
Chapter 2	References used in this specification
Chapter 3	Abbreviations used in this specification
Chapter 4	DSRC-specification. Defines the DSRC-link technical requirements.
Chapter 5	EFC-transaction. Defines the EFC-transaction in terms of transaction structure, and the sequencing of commands and application data.
Chapter 6	Security. Defines the technical elements of the security provisions.
Chapter 7	Data elements. Defines the data elements to be used by the OBU (including classification data).
Annex A	EFC architecture. Provides the overall context of the EFC service
Annex B	EFC DSRC transactions. Accounts for migration path considerations and compares the harmonised PISTA and CARDME transactions.
Annex C	Vehicle classification concepts. Elaborates on the vehicle classification concepts and the associated vehicle data.
Annex D	Data specification. Defines the data attributes.
Annex E	Security implementation examples. Illustrates the defined cryptographic mechanisms by means of a few numerical examples.

## 2 References

This specification incorporates by dated or undated reference, provisions from other publications.

These references are cited, including the relevant chapter(s) when applicable, at the appropriate places 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 specification only when incorporated in it by amendment or revision. For undated references the latest edition of the publication referred to applies.

Reference	Document no	Date	Document title
[EN L1]	prEN 12253 <sup>2,3</sup>	2003	Road Transport and Traffic Telematics (RTTT) – Dedicated Short-Range Communication (DSRC) – Physical layer using microwave at 5.8 GHz
[EN L2]	EN 12795	2002	Road Transport and Traffic Telematics (RTTT) – Dedicated Short-Range Communication (DSRC) – Medium access and logical link control
[EN L7]	EN 12834	2002	Road Transport and Traffic Telematics (RTTT) – Dedicated Short-Range Communication (DSRC) – Application Layer
[EN Profiles]	prEN 13372 <sup>3,4</sup>	2003	Road Transport and Traffic Telematics (RTTT) – Dedicated Short-Range Communication (DSRC) – DSRC profiles for RTTT applications
[AVI No]	ENV ISO 14816 <sup>3</sup>	1999	Road Traffic and Transport Telematics (RTTT) – Automatic Vehicle and Equipment Identification – Numbering and Data Structures
[EFC AID]	prEN ISO 14906 <sup>3,5</sup>	2003	Road Traffic and Transport Telematics (RTTT) – Electronic Fee Collection – Application interface definition for dedicated short range communication
[ETSI DSRC]	EN 300 674		Electromagnetic Compatibility and Radio Spectrum Matters (ERM) - Technical characteristics and test methods for DSRC transmission equipment (500 kbit/s / 250 kbit/s) operating in the 5,8 GHz ISM band
[R&TTE]	Directive 1999/5/EC	1999-03-09	Directive 1999/5/EC of the European Parliament and of the Council of 9 March 1999 on radio equipment and telecommunications terminal equipment and the mutual recognition of their conformity
[ISO 7812-1]	EN ISO/IEC 7812-1	2000	Identification cards - Identification of issuers - Part 1: Numbering system
[DEA]	ISO 8731-1	1987	Banking—Approved algorithms for message authentication—Part 1: DEA
[ASN.1]	ISO/IEC 8824-1	1998	Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation

<sup>2</sup> This document is compliant with prEN 12253 ("DSRC Physical Layer", CEN/TC278 N1549).

<sup>3</sup> The ambition is to adopt the corresponding EN/(ISO) version, which is under preparation.

<sup>4</sup> This document is compliant with prEN 13372 ("DSRC Profiles ", CEN/TC278 N1550).

<sup>5</sup> CEN prEN / ISO DIS 14906 is being launched for parallel Formal Vote.

Reference	Document no	Date	Document title
[ASN.1 PER]	ISO/IEC 8825-2	1998	Information technology - ASN.1 encoding rules: Specification of Packed Encoding Rules (PER)
[ISO 612]	ISO 612		Road vehicles - Dimensions of motor vehicles and towed vehicles - Terms and definitions
[ISO 1176]	ISO 1176		Road vehicles - Masses; vocabulary and codes
[Draft EC Directive]	COM(2003) 132 Final 2003/0081 (COD)	2003-04-23	Proposal for a Directive of the European Parliament and of the Council on the widespread introduction and interoperability of electronic road toll systems in the Community
[MÅNS]		2001-03-01	MÅNS Project – Final reports (2003-03-01) <ul style="list-style-type: none"> <li>• Guidelines on manual payment (1.1, v. 1.75)</li> <li>• Guidelines on automatic payment using central accounts (2.1, v. 1.31)</li> <li>• Guidelines on classification (3.1, v. 1.9)</li> <li>• Guidelines on user information (4.1, v. 1.4)</li> <li>• Guidelines on exception handling (5.1, v. 5.1)</li> <li>• Guidelines on clearing of EFC transactions (6.1, v. 1.4)</li> <li>• Proposal on contractual interoperability (MoU) between Denmark, Finland, Norway and Sweden (7.1, v. 1.5)</li> <li>• The MÅNS terminology (8.1, v. 1.4)</li> <li>• The MÅNS approach (8.2, v. 1.3)</li> <li>• The MÅNS policy (8.3, v 1.1)</li> </ul>
[CARDME]	IST-1999-29053 Deliverable 4.1	2002	CARDME-4 – The CARDME concept (Final, 1 June 2002)
[CESARE]	D.032.1	2002-02-27	CESARE II Project. Detailed CESARE Technical Specification.
[PISTA]	IST-2000-28597 D3.4	2002-11-11	PISTA – Transaction Model
[GSS]		2003	Global Specification for Short Range Communication (Kapsch TrafficCom AB, Kapsch Telecom GmbH, Thales e-Transactions CGA SA, version 3.2, 2003-08, <a href="http://www.etc-interop.com/pdf/gss_32.pdf">http://www.etc-interop.com/pdf/gss_32.pdf</a> )

### 3 Abbreviations

For the purpose of this specification, the following abbreviations apply throughout the document unless otherwise specified:

AID	Application Interface Definition
ASECAP	European Association of Companies with Concessions for Motorway, Bridge and Tunnel Tolls (Association européenne des Concessionnaires d'Autoroutes et des Ouvrages à Péage, <a href="http://www.asecap.com">www.asecap.com</a> )
ATB	General Technical Specification (Allmän Teknisk Beskrivning)
BST	Beacon Service Table
CARDME	Concerted Action for Research on Demand Management in Europe ( <a href="http://www.cardme.org">www.cardme.org</a> )
CEN	European Committee for Standardization (Comité Européen de Normalisation, <a href="http://www.cenorm.be">www.cenorm.be</a> )
CESARE	Common EFC System for ASECAP Road Tolling European System
DEA	Data Encryption Algorithm
DSRC	Dedicated Short-Range Communication
EC	European Commission
EID	Element Identifier
EFC	Electronic Fee Collection
EN	European (CEN) Standard
ETSI	European Telecommunications Standards Institute ( <a href="http://www.etsi.org">www.etsi.org</a> )
GSS	Global Specification for Short Range Communication
HGV	Heavy Goods Vehicle
ISO	International Organization for Standardization ( <a href="http://www.iso.ch">www.iso.ch</a> )
L1	Layer 1 of DSRC (Physical Layer)
L2	Layer 2 of DSRC (Data Link Layer)
L7	Layer 7 of DSRC (Application Layer)
LLC	Logical Link Control
MAC	Message Authentication Code
MMI	Man-Machine Interface
MoU	Memorandum of Understanding
MÅNS	Achieving Interoperability between the Nordic Payment Payment Means for Road User Charges
N/A	Not Applicable
OBU	On-Board Unit
PER	Packed Encoding Rules (ISO/IEC 8825-2)
PISTA	Pilot on Interoperable Systems for Tolling Applications
PSP	Payment Service Provider
PW	Password
RSE	Roadside Equipment
SNRA	Swedish National Road Administration ( <a href="http://www.vv.se">www.vv.se</a> )
T-APDU	Transfer-Application Protocol Data Unit
TSP	Transport Service Provider
TTP	Trusted Third Party
VST	Vehicle Service Table

## 4 DSRC requirements

This chapter defines the DSRC requirements related to communication between the OBU and RSE.

The OBU shall comply with DSRC Profiles P0 / P1 and parameter set L1-B [EN Profiles]. The RSE shall support the full spectrum of possible OBU DSRC implementations complying with DSRC Profiles P0 / P1, parameter set L1-B [EN Profiles] and the additional precision below in this chapter.

NOTE: This implies that the OBU and RSE shall comply with the underlying DSRC-L1 [EN L1], -L2 [EN L2], -L7 [EN L7].

The OBU and RSE shall comply with the European [R&TTE] Directive, which means that conformance testing of DSRC-L1 [EN L1] shall be performed according to [ETSI DSRC].

The OBU (low-level) behaviour shall comply with the behaviour defined by the state transition tables in chapter 6.3 in [GSS]. The RSE shall support OBU exhibiting the (low-level) behaviour as defined by the state transition tables in chapter 6.3 in [GSS].

The OBU and RSE shall comply with the DSRC-L2 [EN L2] including the defined data link parameters in its Annex A.

The OBU and RSE shall comply with the DSRC-L7 [EN L7]. The following DSRC-L7 features shall be supported by the OBU:

- Concatenation of multiple consecutive T-APDU fragments in one layer 2 frame (i.e. LLC-service) with and without chaining, given that the size constraint for the LLC-frames are not violated (i.e. fit into 1 L2 frame);
- Fragmentation header equalling 1 octet only;
- Any "fill bit" (as defined 6.3.4 in EN L7), used for octet alignment, shall be assigned the value zero.

The following DSRC-L7 features need not be supported by the OBU and the RSE:

- Fragmentation and defragmentation;
- Multiplexing and demultiplexing;

The OBU and RSE shall support downlink frames 1-2 and 4-7 (i.e. all frames except downlink frame no 3) as defined in Table 4 in [EN Profiles].

The OBU and RSE shall support uplink frames 1-6 as defined in Table 5 in [EN Profiles].

## 5 EFC transaction requirements

This chapter defines the requirements related to the EFC transaction between the OBU and the RSE. The details of the security features and application data deployed in the EFC transaction are further defined in chapters 6 and 7, respectively.

### 5.1 EFC transactions

The EFC transaction model complies with [EFC AID, chapter 6].

The OBU shall comply with either the [PISTA] or the [CARDME] transaction and the options defined in 5.1.1 and 5.1.2, respectively.

The RSE shall support the [PISTA] transaction and the options defined in defined in 5.1.1. The RSE should also support the [CARDME] transaction and the options defined in 5.1.2.

NOTE: The RSE knows which EFC transaction(s) is/are supported by the OBU when it has evaluated the VST, and it can then select which transaction to execute with a certain OBU.

The tables in 5.1.1 and 5.1.2 specify the EFC transactions in terms of the sequence of (DSRC-L7 and EFC function) service primitives and application data exchanged.

Attributes marked with an \* in Table 5.1 and Table 5.2 below are optional and can be retrieved at the discretion of the RSE. See chapter 7 for details of requirements with respect to attributes.

### 5.1.1 PISTA transaction

The table below specifies the [PISTA] transaction in terms of the sequence of (DSRC-L7 and EFC function) service primitives and application data exchanged.

Phase	Roadside Equipment		On-board unit	Remarks
Initialisation & contract selection (BST – VST)	INITIALISATION.request (BST)	→		
		←	INITIALISATION.response (VST) <ul style="list-style-type: none"> <li>EFC-ContextMark</li> <li>EquipmentClass</li> <li>ManufacturerId</li> <li>OBUStatus</li> </ul>	
Presentation (data retrieval & evaluation)	GET_STAMPED.request <sup>6</sup> <ul style="list-style-type: none"> <li>ContractAuthenticator (RndRSE, KeyRef)</li> </ul> GET.request <sup>7</sup> <ul style="list-style-type: none"> <li>VehicleClass *</li> <li>VehicleDimensions *</li> <li>VehicleAxes *</li> <li>VehicleAuthenticator *</li> <li>EquipmentOBUID *</li> <li>PaymentMeans (including PersonalAccountNumber)</li> <li>ReceiptData1 (last)</li> <li>ReceiptData2 (penultimate) *</li> </ul>	→		
		←	GET_STAMPED.response GET.response	
Receipt Generation	SET.request <ul style="list-style-type: none"> <li>ReceiptData1</li> <li>ReceiptData2</li> </ul> SET_MMI.request	→		
		←	SET.response SET_MMI.response	
Tracking And Closing	ECHO.request	→		
		←	ECHO.response	
	EVENT-REPORT.request (Release)	→		

Table 5.1 PISTA transaction (security level 2)

<sup>6</sup> GET is used in CESARE / PISTA level 1 to retrieve ContractAuthenticator.

<sup>7</sup> In case of retrieval of all the optional attributes, two GETs may be necessary.

## 5.1.2 CARDME transaction

The table below specifies the [CARDME] transaction in terms of the sequence of (DSRC-L7 and EFC function) service primitives and application data exchanged.

Phase	Roadside Equipment		On-board unit	Remarks
Initialisation	<b>INITIALISATION.request (BST)</b>	→		RSE periodically sends BST.
		←	<b>INITIALISATION.response (VST)</b> <ul style="list-style-type: none"> <li>EFC-ContextMark</li> <li>AC_CR-KeyReference</li> <li>RndOBU</li> </ul>	A newly arrived OBU answers with a standardised VST: <ul style="list-style-type: none"> <li>EFC-ContextMark consists of ContractProvider, TypeOfContract and ContextVersion.</li> <li>Gives the reference for the Access Credential Keys to use by the RSE</li> <li>RndOBU is a Random Number that the RSE uses for calculation of Access Credentials</li> </ul>
Presentation	<b>GET_STAMPED.request</b> AC_CR <ul style="list-style-type: none"> <li>PaymentMeans, including PersonalAccountNumber (RndRSE, KeyRef_Op)</li> </ul> <b>GET.request</b> AC_CR = PW <ul style="list-style-type: none"> <li>ContractAuthenticator*</li> <li>Vehicle data:               <ul style="list-style-type: none"> <li>VehicleLicencePlateNumber*</li> <li>VehicleClass*</li> <li>VehicleDimensions*</li> <li>VehicleAxles*</li> <li>VehicleWeightLimits*</li> <li>VehicleSpecificCharacteristics*</li> <li>VehicleAuthenticator*</li> </ul> </li> <li>EquipmentStatus (transaction counter)</li> <li>ReceiptData1</li> <li>ReceiptData2*</li> </ul>	→		Request OBU to provide the Personal Account Number (pointing to the user's contract/account at the contract issuer) with an Authenticator. OBU will give access only when RSE provides the correct Access Credentials AC_CR. Random number and key reference for the authenticator that the OBU is to calculate. Request of the data that are used for user data verification and fee determination purposes <ul style="list-style-type: none"> <li>ContractAuthenticator</li> <li>declared vehicle data               <ul style="list-style-type: none"> <li>vehicle license plate number</li> <li>vehicle class also gives information on trailer presence</li> <li>vehicle axles includes information on presence of dual tyres</li> <li>vehicle specific characteristics include information on emission class, engine type, etc.</li> </ul> </li> <li>VehicleAuthenticator</li> <li>equipment status (which includes a transaction counter)</li> <li>last receipt (entry ticket or last transaction)</li> <li>penultimate (i.e. second last) receipt</li> </ul>
		←	<b>GET_STAMPED.response</b> <ul style="list-style-type: none"> <li>Operator_Authenticator (Auth_Op)</li> </ul> <b>GET.response</b>	OBU responds with the data asked for, including the Authenticator calculated with the 'interoperable key', i.e. with a key known to all EFC Operators. The Authenticator provides data integrity of the payment means and OBU authenticity
Foreign OBU presentation (optional)	<b>GET_STAMPED.request</b> AC_CR <ul style="list-style-type: none"> <li>PaymentMeans, including PersonalAccountNumber (RndRSE, KeyRef_Iss)</li> </ul>	→		The RSE asks for the calculation of an additional authenticator, for OBUs from a foreign Contract Issuer, over the Payment Means (incl. Personal Account Number) with keys only known to the Contract Issuer, so that one can prove that the vehicle actually passed the charging point.
		←	<b>GET_STAMPED.response</b> <ul style="list-style-type: none"> <li>Issuer_Authenticator (Auth_Iss)</li> </ul>	
Receipt	<b>SET.request</b> AC_CR <ul style="list-style-type: none"> <li>EquipmentStatus (transaction counter)</li> <li>ReceiptData1</li> <li>ReceiptData2</li> </ul> <b>SET_MMI.request</b>	→		Write new status information and increment transaction counter. Write new receipts (or entry ticket)  Give an 'OK' indication to the user (normally the OBU will beep)
		←	<b>SET.response</b> <b>SET_MMI.response</b>	
Tracking and Closing	<b>ECHO.request</b>	→		Track OBU by exchanging dummy information.
		←	<b>ECHO.response</b>	The usage of Echo is optional, at the discretion of the RSE, and may be repeated.
	<b>EVENT_REPORT.request (Release)</b>	→		RSE closes the transaction and releases the OBU

Table 5.2 CARDME transaction

## 5.2 DSRC L7 services and EFC functions

The OBU and the RSE shall support the DSRC Layer 7 services and EFC functions, defined in [EN L7] and [EFC AID, chapter 7.2], in Table 5.3 that are necessary for the execution of the EFC transaction.

DSRC-L7 services	EFC function	Action / Event type	Remarks
INITIALISATION	N/A	N/A	Establishes communication, selects the application and contract
ACTION	GET_STAMPED	0	Retrieves data with an authenticator from the OBU
GET	N/A	N/A	Retrieves data from the OBU
SET	N/A	N/A	Writes data to the OBU
ACTION	SET_MMI	10	Invokes an MMI function (e.g. signal "Ok" via buzzer). All SetMMIRq values (i.e. 0, 1, 2 and 255) defined in Annex A in [EFC AID] shall be supported.
ACTION	ECHO	15	OBU echoes received data
EVENT-REPORT	RELEASE	1	Terminates communication

**Table 5.3** DSRC L7 services and EFC functions

The addressing of the EFC system and application data shall conform to the rules defined in 5.3 in [EFC AID]. No access credentials apply in case of a PISTA transaction, whereas access credentials apply in case of a CARDME transaction for the GET\_STAMPED, GET and SET functions.

## 6 Security features

This chapter contains a description of the basic requirements on the security features in the system. Annex E illustrates the defined cryptographic mechanisms by means of a few numerical examples.

### 6.1 Security features overview

The EFC transactions encompass the following security features:

- **Authenticator:** providing data integrity and data origin authenticity of the “payment means” (i.e. challenge-response of ContractAuthenticator or PaymentMeans data, using the GET\_STAMPED function).
- **Access credentials:** data access protection to OBU, i.e. verification of RSE authenticity and protection against non-authorised access of OBU data.
- **Transaction counter (according to 3.2.4 in Parts 1-2 in CARDME):** increased by and at the discretion of the RSE, allowing detection of transaction sequencing anomalies in the central system.
- **Signed receipt (i.e. ReceiptAuthenticator in ReceiptData1/2):** generated by and at the discretion of the RSE ensuring data integrity of the Receipt data.
- **VehicleAuthenticator (according to PISTA):** carries the result of a cryptographic calculation, done by the issuer, using all the vehicle attributes. VehicleAuthenticator is calculated with a “common algorithm” created and distributed with the MoU, since all operators need to be able to check them. It is calculated by the issuer upon personalisation of the OBU with an algorithm known by the operators and the MoU. Each operator may decide whether to retrieve this attribute or not during the transaction, and therefore whether checking it or not, eventually in real time or in post processing.
- **ContractAuthenticator (according to PISTA):** carries the result of a cryptographic calculation, done by the issuer and unknown to the operator, using the EFCContextMark and PaymentMeans. Each issuer must define a cryptographic algorithm to calculate the authenticator, compulsory to initialise this authenticator by the issuer. Used for post/processing checking of the information integrity by the issuer.

The OBU shall be able to generate an Authenticator<sup>8</sup> (i.e. support the GET\_STAMPED function operating on a single attribute). The OBU may in addition also support Access credentials, needed in case of [CARDME] transactions.

NOTE: The OBU also supports (through the data stored in the OBU) the other security features (i.e. Transaction counter, Signed receipt, VehicleAuthenticator and ContractAuthenticator) that are performed at the RSE or in the central system.

The RSE security features shall encompass Authenticator, and should encompass Access credentials and Signed receipt. The RSE or the Central System should encompass verification of the VehicleAuthenticator and, in case of the Issuer also, ContractAuthenticator.

The commonly defined cryptographic real-time security features – Authenticator and Access Credentials – are defined in the 6.2 and 6.3 below.

---

<sup>8</sup> Not used in the PISTA level 1 transaction, but this GET\_STAMPED functionality is mandatory also for “PISTA level 1 OBU” in order to ensure seamless migration to PISTA level 2.

## 6.2 Computation of authenticator

The computation of authenticator (both Auth\_Issuer and Auth\_Operator generated using the Issuer and Operator keys, respectively) shall be performed according to [DEA].

### 6.2.1 ContractAuthenticator authenticator

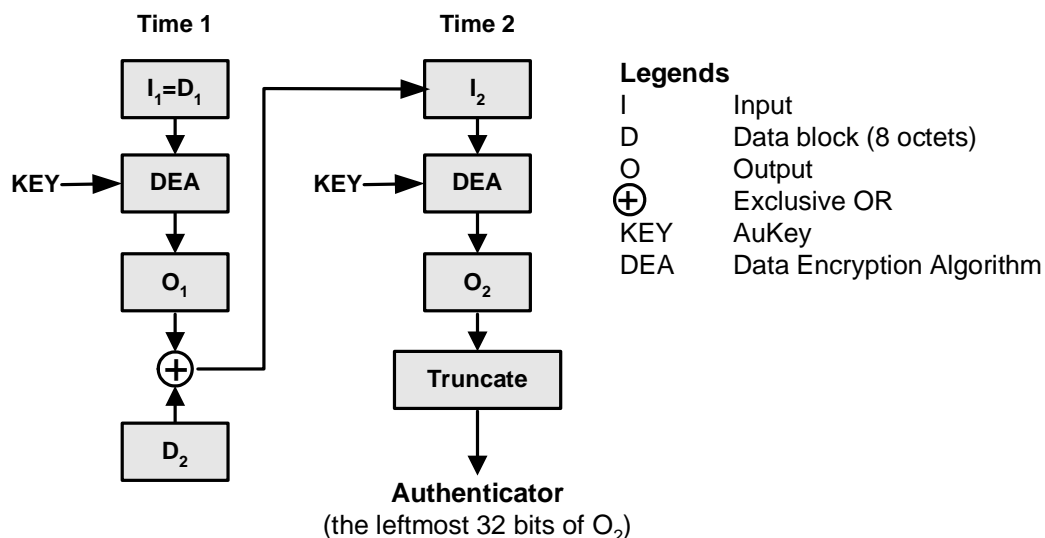
The following procedure shall be used to compute the Authenticator – ContractAuthenticator - used in [PISTA] level 2 transactions for a given key generation (i):

- Let M be the Attribute List in the GET\_STAMPED response containing the single attribute ContractAuthenticator, concatenated by the octet string containing the RndRSE sent by the RSE in the GET\_STAMPED request. M will now have the following content:

Octet #	Attribute / Field	Bits in Octet b <sub>7</sub> b <sub>0</sub>	Description
1	AttributeList SEQUENCE (0..127,...) OF {	0000 0001	No extension, number of attributes: 1
2	Attributes SEQUENCE { AttributeId	0000 0100	ContractAuthenticator = 4 <sub>10</sub>
3	AttributeValue CONTAINER {	0010 0100	Container Choice: 36 <sub>10</sub> = ContractAuthenticator
4	ContractAuthenticator	xxxx xxxx	ContractAuthenticator
5		xxxx xxxx	
6		xxxx xxxx	
7		xxxx xxxx	
8		xxxx xxxx	
9	Nonce OCTET STRING {	0000 0100	No extension, octet string length = 4 <sub>10</sub>
10	RndRSE	ffff ffff	Random number from RSE, containing the SessionTime
11		ffff ffff	
12		ffff ffff	
13		ffff ffff	
14	Padding	0000 0000	Padding to obtain even 8-octet blocks
15	Padding	0000 0000	Padding to obtain even 8-octet blocks
16	Padding	0000 0000	Padding to obtain even 8-octet blocks

**Table 6.1** Content of M – the message that is used as input to generate the authenticator

- Let D1 be the first 8 octets, D2 be 9-16 octets and let D3 be the last 8 octets of the message M.
- Let Key in the figure below be the AuthenticationKey(i) where (i) is the KeyRef sent by the RSE.
- Compute the Authenticator according to the DES algorithm [DEA].



**Figure 6.1** Illustration of the computation of ContractAuthenticator authenticator.

## 6.2.2 PaymentMeans authenticator

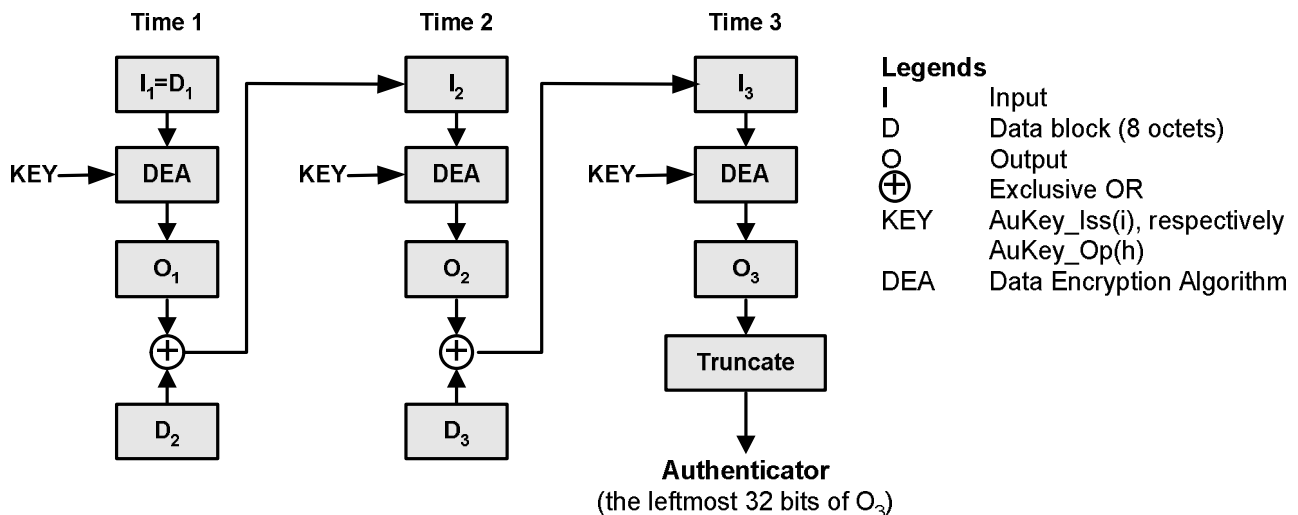
The following procedure shall be used to compute the Authenticator – PaymentMeans - used in [CARDME] transactions for a given key generation (i):

- Let M be the Attribute List in the GET\_STAMPED response containing the single attribute PaymentMeans (including Personal Account Number), concatenated by the octet string containing the RndRSE sent by the RSE in the GET\_STAMPED request. M will now have the following content:

Octet #	Attribute / Field	Bits in Octet b <sub>7</sub> b <sub>0</sub>	Description
1	AttributeList SEQUENCE (0..127,...) OF {	0000 0001	No extension, number of attributes: 1
2	Attributes SEQUENCE { AttributeId	0010 0000	PaymentMeans = 32 <sub>10</sub>
3	AttributeValue CONTAINER {	0100 0000	Container Choice: 64 <sub>10</sub> = PaymentMeans
4	PaymentMeans	xxxx xxxx	PaymentMeans
5		xxxx xxxx	
6		xxxx xxxx	
7		xxxx xxxx	
8		xxxx xxxx	
9		xxxx xxxx	
10		xxxx xxxx	
11		xxxx xxxx	
12		xxxx xxxx	
13		xxxx xxxx	
14	PaymentMeansExpiryDate	0001 1110	DateCompact. Example : 2005-03-01
15	PaymentMeansUsageControl	0110 0001	Example : No specific restrictions (0)
16		0000 0000	
17		0000 0000	
18	Nonce OCTET STRING {	0000 0100	No extension, octet string length = 4 <sub>10</sub>
19	RndRSE	rrrr rrrr	Random number from RSE, containing the SessionTime
20		rrrr rrrr	
21		rrrr rrrr	
22		rrrr rrrr	
23	Padding	0000 0000	Padding to obtain even 8-octet blocks
24		0000 0000	

**Table 6.2** Content of M – the message that is used as input to generate the authenticator

- Let D1 be the first 8 octets, D2 be 9-16 octets and let D3 be the last 8 octets of the message M.
- Let Key in the figure below be the AuthenticationKey(i) where (i) is the KeyRef sent by the RSE.
- Compute the (Issuer or Operator) Authenticator according to the DES algorithm [DEA].

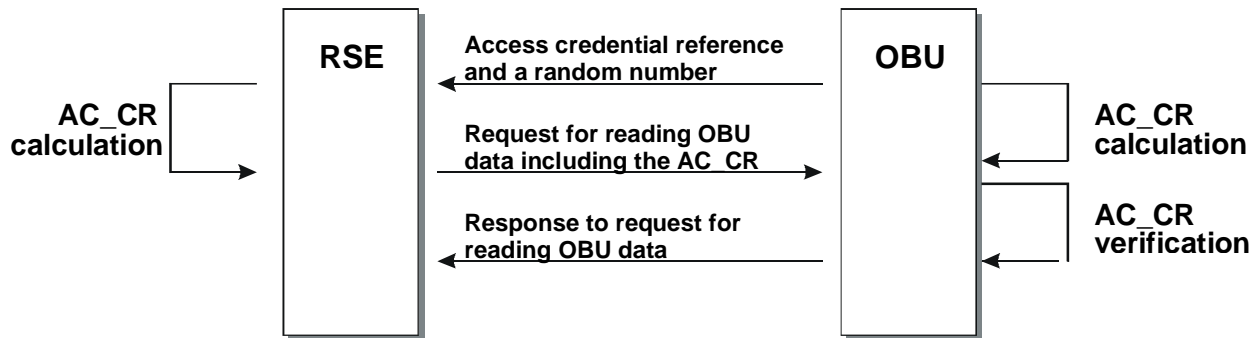


**Figure 6.2** Illustration of the computation of PaymentMeans authenticator.

### 6.3 Access credentials

Access credentials are used in [CARDME] transactions, in order to protect against non-authorised access to sensitive user data and against (commercial) use of the OBU by non-authorised operators.

The principle of access control to the OBU information is shown below. When an OBU, having entered the communication zone, responds to a polling message (BST) from the RSE, it returns a VST that for each available Contract contains information about an Access Credential Reference (AC\_CR-Reference) and a random number (RndOBU). The AC\_CR\_Reference includes the data AC\_CR-MasterKeyReference and AC\_CR-Diversifier. The data are the diversifier and a reference to a secret key (MasterKey for Access Credentials) that shall be used for the computation of the secret key (AC\_CRKey). This key shall be used for the computation of the Access credentials (AC\_CR) using the RndOBU number as input. The RSE returns the access credential calculated and the OBU compares the access credential with its own calculation. In case they are equal the OBU accepts the RSE as a genuine RSE and reading data from the OBU is allowed.



**Figure 6.3** The principle of access control to the OBU data.

The OBU and RSE computation of Access credentials (AC\_CR) based on DES are defined in the two subsequent chapters, respectively.

### 6.3.1 OBU computation of Access credentials

The OBU shall use the following procedure for computation of the Access credentials (AC\_CR) for a given key reference (k):

1. Get the RndOBU
2. Make the concatenation of 'RndOBU || 00 00 00 00' to obtain an 8 octets value VAL:  
VAL = 'RndOBU || 00 00 00 00'
3. Let VAL be D1
4. Compute the AC\_CR(k) according to the DES algorithm [DEA]:

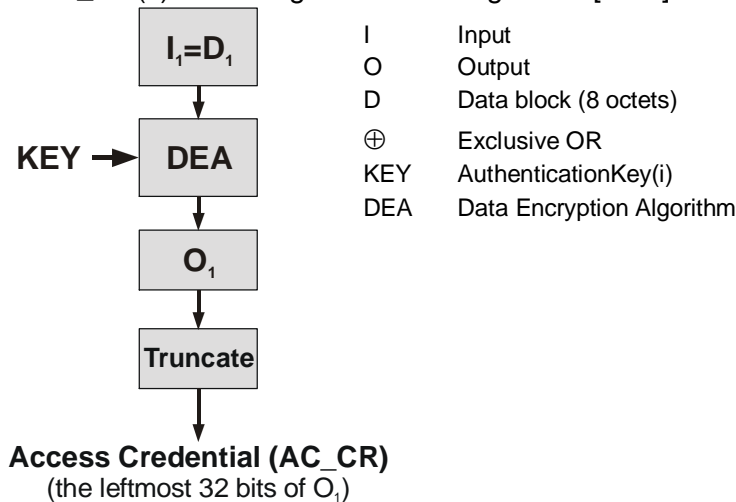


Figure 6.4 Computation of Access Credentials.

### 6.3.2 RSE computation of Access credentials

The RSE shall use the following procedure for computation of the Access credentials (AC\_CR) for a given key reference (k):

1. Set k = AC\_CR-MasterkeyReference (sent by the OBU in VST)
2. Get the MAC\_CRKey(k) (stored in the RSE)
3. Get the AC\_CR-Reference (2 octets sent by the OBU in VST)
4. Make the concatenation of 'AC\_CR-Reference || AC\_CR-Reference || AC\_CR-Reference || AC\_CR-Reference' to obtain an 8 octets value VAL1:  
VAL1 = 'AC\_CR-Reference || AC\_CR-Reference || AC\_CR-Reference || AC\_CR-Reference'
5. Compute the AC\_CRKey(k) as follows:  
AC\_CRKey(k) = ede[MAC\_CRKey(k)] (VAL1)
6. Get the RndOBU sent by the OBU in VST
7. Make the concatenation of 'RndOBU || 00 00 00 00' to obtain an 8 octets value VAL2 = 'RndOBU || 00 00 00 00'
8. Let VAL2 be D1
9. Compute the AC\_CR(k) according to the DES algorithm [DEA]

## 7 Data attribute requirements

This chapter defines the application data attribute requirements based on [EFC AID]. Annex D specifies the application data in further details, including authentication keys, access keys and arguments in security related EFC functions (see Table D.7).

### 7.1 Overview

Below an overview of the application data attribute requirements based on [EFC AID]. All attributes are mandatory in the OBU. The personalisation requirements are accounted for in 7.2.

The following columns are used in the table below:

- data group and attribute name;
- attribute id;
- length in octets (PER encoded);
- type, i.e. the Container choice type value;
- read: denotes whether the data is subject to retrieval (i.e. “reading”) or not during an EFC transaction<sup>9</sup>, at the discretion of the RSE;
- write: denotes whether the data is subject to setting (i.e. “writing”) or not during an EFC transaction;
- Remarks: miscellaneous highlights.

ATTRIBUTES (EID>0)	AttrId	Type	Length	Read	Write	Remarks
CONTRACT						Information associated with the service rights of the issuer of the EFC service.
EFC Context Mark	0	32	6	Yes	No	Contains the Contract Provider. Transmitted as part of the VST.
ContractAuthenticator	4	36	5	Yes	No	
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.
VEHICLE						Information pertaining to the identification and characteristics of the vehicle.
VehicleLicencePlateNumber	16	47	13	Yes	No	
VehicleClass	17	49	1	Yes	No	
VehicleDimensions	18	50	3	Yes	No	
VehicleAxles	19	51	2	Yes	No	
VehicleWeightLimits	20	52	6	Yes	No	
VehicleSpecificCharacteristics	22	54	4	Yes	No	
VehicleAuthenticator	23	55	5	Yes	No	
EQUIPMENT						Information pertaining to the OBU.
EquipmentOBUId	24	56	5 (=1+4)	Yes	No	
EquipmentStatus (transaction counter)	26	58	2	Yes	Yes	
RECEIPT						Information associated with a specific session, including both financial and operational data.
ReceiptData1 (last)	33	65	28	Yes	Yes	
ReceiptData2 (penultimate)	34	66	28	Yes	Yes	

**Table 7.1** Overview of the OBU EFC application data

The reading and writing of the EFC transaction data, except for the EFC Context Mark, is subject to access conditions in CARDME transactions.

<sup>9</sup> Personalisation of the OBU is outside the scope of this specification.

## 7.2 Personalisation requirements

All attributes shall be personalised (be it with "dummy" data) and be available in the OBU.

Three OBU personalisation concepts, associated with sub-contract types, are distinguished according to Table 7.2 below. See Annex C for the vehicle classification concepts.

Legend:

- Entry: personalisation of the OBU with the data according characteristics of the vehicle
- Dummy: personalisation of the OBU with "no entry".

Vehicle data	Sub-contract <sup>10</sup>		
	Type 1 Full vehicle data	Type 2 Pre-defined class	Type 3 No vehicle data
VehicleClass	Entry	Entry	Dummy
VehicleLicencePlateNumber	Entry	Dummy	Dummy
VehicleDimensions	Entry	Dummy	Dummy
VehicleAxles	Entry	Dummy	Dummy
VehicleWeightLimits	Entry	Dummy	Dummy
VehicleSpecificCharacteristics	Entry	Dummy	Dummy

**Table 7.2** OBU personalisation requirements

<sup>10</sup> Part of TypeOfContract. See EFC-ContextMark in Annex D.

## ANNEX A: EFC ARCHITECTURE

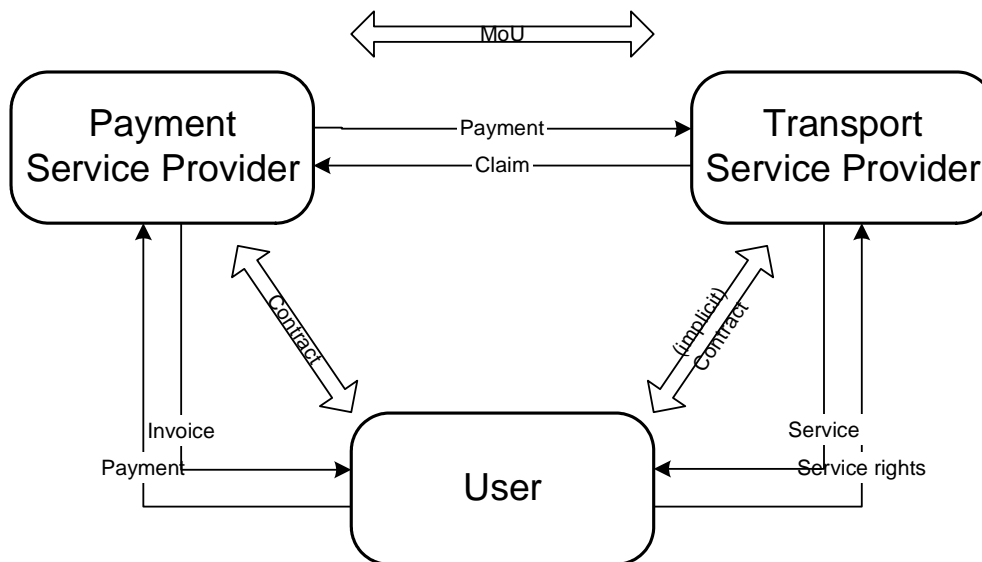
This specification defines basic requirements on interoperable EFC, within the (rather narrow) scope as defined in Chapter 1. As the specification covers only parts (sub-systems and interfaces) of a full EFC service (based on central account), the overall picture of the EFC service needs to be given to understand the context and scope of the specification. This annex aims at defining the generic architecture for such an EFC-service for interoperable use. Although not specifically mentioned in the document, this also puts some requirements on other parts of the system. E.g. the data element definition (for the transaction) also defines a “short-list” of data to be used for claims (between PSP and TSP). E.g. the security mechanisms also imply the handling of security keys between different PSPs and TSPs.

The description is divided into three perspectives on the architecture for the service<sup>11</sup>:

- Actors' perspective;
- Functional perspective;
- Physical perspective.

### A.1 Actors' perspective

The basic inter-organisational entities in interoperable EFC have been defined in several projects, where MÅNS, CESARE and CARDME are the most important ones. Although the terminology slightly differs between them, the basic concepts are the same. The Swedish Actors architecture follows the result of these projects, and is given in the figure below:



**Figure A. 1** The Swedish EFC architecture (actors' perspective)

This actor model shows the basic (minimum) roles and responsibilities in the interoperable EFC service. In actual operation of the EFC service these responsibilities might be further decomposed, combined in organisations, or delegated to other entities. The figure shows the core actors, and their main responsibilities are defined below. In addition a full and unambiguous agreement of

<sup>11</sup> This notation corresponds to the Swedish ITS Architecture currently being developed, and is in line with the European framework for ITS architectures developed in the KAREN-project.

those must be defined by a MoU between the actors. This is not within the scope of this specification.

### **A.1.1 User**

This is the User of the payment service and the transport service offered. The User enters a contract for the use of a payment means (an OBU, central account, service rights) with a payment Service provider. This contract enables the User to use the payment means for payment of fee at several Transport Service Providers within the MoU. The User is invoiced by the Payment Service Provider and pays money to the Payment Service Provider. When the User uses a service offered by the Transport Service Provider an implicit contract is entered for the use of the service (e.g. under some general conditions).

### **A.1.2 Payment service provider**

The Payment Service Provider (also called Issuer) is the entity responsible for the payment means (an OBU, central account, service rights). The Payment Service Providers relations to Users are described above. The Payment Service Provider also enters a MoU together with other Payment or Transport Service Provider, which regulates the use of the interoperable EFC service (contracts, procedures, clearing, technology, etc).

### **A.1.3 Transport service provider**

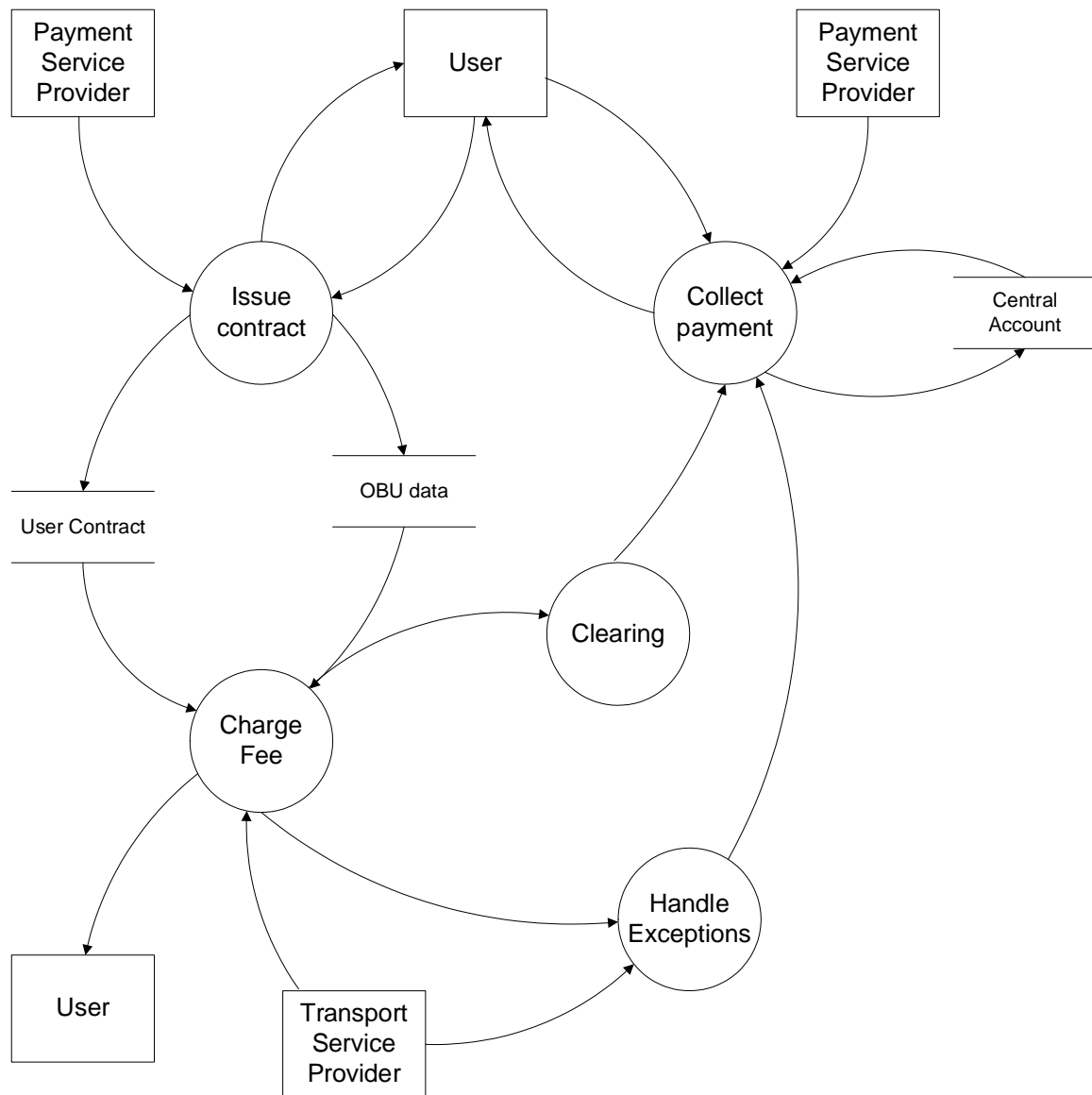
The Transport Service Provider (also called EFC Operator) is the entity offering a transport service to the User (e.g. toll road access). The Transport Service Provider performs a transaction when the service is being charged and compiles claims for money from the Payment Service Provider to which the Users contract belongs. The Transport Service Providers relations to Users and Payment Service providers are described above.

Please note that in many cases (e.g. in the Nordic countries today) the Transport Service Provider is also a Payment Service Provider, i.e. the Operator has its “own” clients. However, this need not be the case. A Transport Service Provider may actually have no “own” clients, and a Payment Service Provider need not offer a Transport Service. This architecture focuses on the roles in an interoperable environment, not the actual realisation of those roles in anyone particular case.

In some cases there may be an active MoU-organisation taking an operational role also for clearing. This, however, is not necessary and the architecture allows also for fully bilateral operations.

## **A.2 Functional perspective**

The functional (or logical) perspective can be described using two parts; a functional architecture and information architecture. In this document we focus on the functional architecture. The functions in interoperable EFC may be described in many different ways. The following is a data flow diagram of the five basic functions of the interoperable EFC service:



**Figure A. 2** Data flow diagram of the basic functions of the interoperable EFC service.

Please note that the details of this functional architecture are given as an example – and should not be interpreted as a fixed list of necessary functions for all EFC services in Sweden. A more elaborated functional architecture needs to be made.

Note that all these functions are necessary in order to perform the EFC service, but it may not be necessary to provide details requirements on every function for enabling interoperable use on the service.

This specification does not use the functional perspective for defining its scope. It can however be noted that most requirements in this specification is for the basic function; Charge Fee.

### A.2.1 Issue contract

This function means that the User enters a contract with the Payment Service Provider for the interoperable use of certain payment means. This includes sub-functions such as:

- Issue OBU contract
- Issue Payment Service contract
- Inform user on contract details and on special "foreign" provisions
- Put information on OBU
- Transfer OBU to User
- Record information after contract issuing (data base)
- Update user information (if needed).

### A.2.2 Charge fee

This function is performed by the Transport Service Provider when the User uses a service. A transaction is performed. This includes sub-functions such as:

- Inform User before charging
- Inform User during charging
- Inform User after charging
- Detect vehicle
- Perform transaction (incl. Classify vehicle, Communicate with the OBU, Check authentication of OBU, Read OBU data, Check user contract validity, Choose the contract to use for charging, Determine charging fee, Check transaction towards black-list.
- Record information after fee collection

### A.2.3 Handle exceptions

This function is performed by the Transport Service Provider when the User uses a service and some "exception" occurs in the transaction that may lead to enforcement. This includes sub-functions such as:

- Verify charging
- Determine exception
- Register vehicle
- Record information after exception handling
- Enforce Vehicle Registration Holder

### A.2.4 Clearing

The Transport Service Provider sends a claim to (each of) the Payment Service Provider for payment when a successful transaction has taken place<sup>12</sup>. This includes sub-functions such as:

- Send claim to operator
- Verify claim and authorise settling of the account

---

<sup>12</sup> Usually this is done in large batches after many transactions.

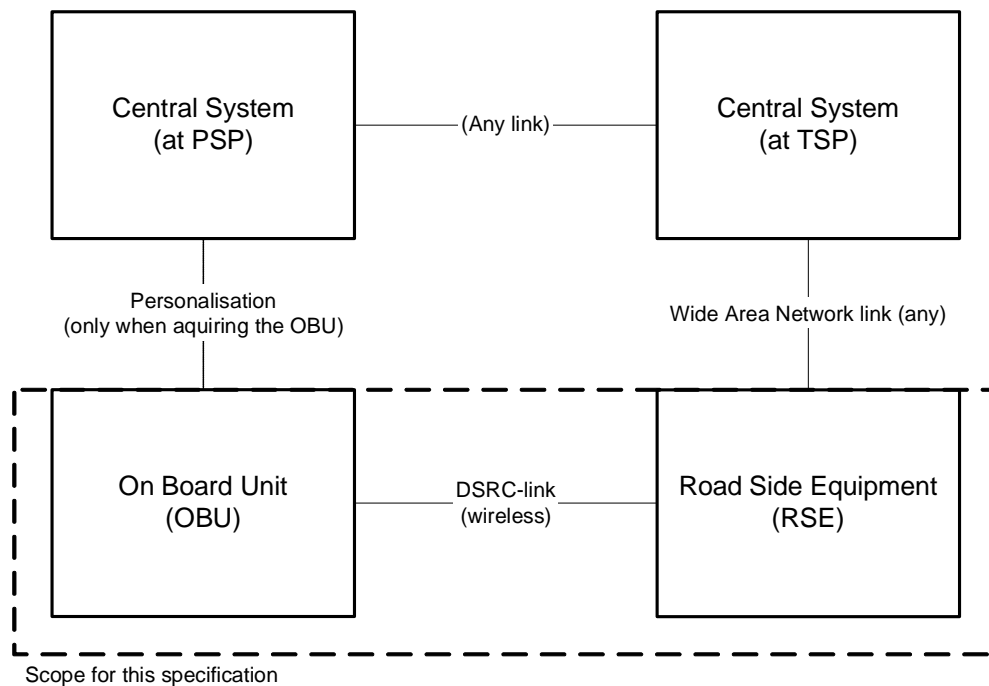
## A.2.5 Collect payment

This function is performed by the Payment Service Provider and includes the actual payment of the fee by the User. Normally this is done by settling the Users central account, held by the Payment Service Provider. This includes sub-functions such as:

- Collect payment for central account (settle account)
- Collect payment for enforcement fee
- Inform user about payments
- Record information after payment collection
- Update user information (if needed)

## A.3 Physical perspective

The physical perspective can be described using two parts; a physical architecture and communications architecture. For the purpose of defining the context and scope in this document we focus on the physical entities<sup>13</sup>. The basic physical entities and interfaces are shown in the following figure:



**Figure A. 3** The basic physical entities and interfaces of the interoperable EFC service.

<sup>13</sup> The EFC transaction described in this specification is largely based on a communications architecture.

## **A.3.1 PHYSICAL ENTITIES**

### **A.3.1.1 On board Unit (OBU)**

This is the basic equipment held by the User in the vehicle. It normally consists of a communications transponder, CPU, memory, buzzer and could (optionally) have a display. For this specification an IC card is not specified (but may be added in a later release).

### **A.3.1.2 Roadside equipment (RSE)**

The RSE is the collective name for all the TSP held equipment at the road-side. The RSE includes DSRC transceivers, and typically detection equipment, classification equipment, video cameras, computers, lighting, gantries, housing, etc.

### **A.3.1.3 Central system (at the Transport Service Provider)**

This is the centrally held equipment at the TSP. It includes mainly off-line computer systems.

### **A.3.1.4 Central system (at the Payment Service Provider)**

This is the centrally held equipment at the TSP. It includes mainly off-line computer systems, but also any equipment for personalisation of OBUs (that may be held at retailers).

## **A.3.2 INTERFACES**

The basic interface in EFC is the DSRC-link between OBU and RSE. This is also the main focus and scope of this specification.

The other interfaces in the above figure (WAN-link, personalisation link) need not given in technical details for interoperability purposes (some requirements or procedures, security, etc. might be added in a MoU).

## ANNEX B – EFC DSRC TRANSACTIONS

### B.1 Migration path considerations

B.1 illustrates migration path considerations for this EFC interoperability specification. These reflect SNRA's perception of the current and projected preferred solution for interoperability of EFC systems in European.

Phase	Features	Remarks
A	"Local" EFC features	Existing local, regional or national EFC system. Common situation for many of today's existing EFC systems.
B	"CESARE / PISTA and CARDME functions" Security: <ul style="list-style-type: none"> <li>"static" authenticators (receipt, vehicle and contract)</li> </ul>	<b>PISTA level 1 [PISTA] - "Static authentication"</b> Gaining popularity among "Motorway and Bridge Toll Concessions". Roaming between a limited numbers of trusted partners. The OBU supports also the GET_STAMPED function from the start in order to ensure seamless migration. No major hardware modifications when migrating from A to B, assuming that the initial system is compliant with CEN DSRC standards. Local/national applications need not be affected, and existing OBU may be phased out gradually. "New" software has to be installed in the RSE to support the "CESARE/PISTA & CARDME" transaction.
C	"CESARE / PISTA and CARDME functions" Security: <ul style="list-style-type: none"> <li>"static" authenticators (receipt, vehicle and contract)</li> <li>dynamic authenticator (Issuer)</li> </ul>	<b>PISTA level 2 [PISTA] - "Dynamic authentication"</b> Logical subsequent step for "Motorway and Bridge Toll Concessions". Roaming between a limited numbers of trusted parties. No major hardware modifications when migrating from A to C, assuming that the initial system is compliant with CEN DSRC standards. Local/national applications need not be affected, and existing OBU may be phased out gradually. "New" software has to be installed in the RSE to support the "CESARE/PISTA & CARDME" transaction.
D	"CESARE / PISTA and CARDME functions" Security: <ul style="list-style-type: none"> <li>"static" authenticators (receipt and vehicle)</li> <li>transaction counter</li> <li>dual dynamic authentication (Issuer and Operator)</li> <li>OBU data access protection</li> </ul>	<b>"Dual authentication and data access protection" [CARDME]</b> Roaming between many EFC operators, including Urban Road User Charging Operators. This solution is associated with dual security domains – roaming and issuer domains – allowing for the Contract Issuer to verify that a transaction claim from a "foreign" Operator indeed is genuine. Further, additional vehicle data are personalised in the OBU (see discussion in Annex C) for "urban / HGV tolling" purposes. Protection against non-authorized access to user data is also provided, primarily in order to safeguard user privacy and integrity. Logical (subsequent) step for a large scale system to introduce stronger protection of user data. No major hardware modifications when migrating from A to D, assuming that the initial system is compliant with CEN DSRC standards. Local/national applications need not be affected, and existing OBU may be phased out gradually. "New" software has to be installed in the RSE to support the "CESARE/PISTA & CARDME" transaction.

**Table B. 1** Migration steps

It should be noted that the RSE can support several transactions, and that OBU generations can co-exist and be phased out gracefully. Interoperable RSE shall support the PISTA transaction and should support the CARDME transaction. The complexity and cost to support both PISTA and CARDME are negligible, as the differences between these transactions are minor, compared with the flexibility to support the two current mainstream transactions in Europe.

A comparison<sup>14</sup> of the PISTA and CARDME transactions is given B.2, showing their similarities and differences. It is intended as support for operators planning (incremental) migration steps.

## B.2 Comparison of PISTA and CARDME

The comparison of the [PISTA] and [CARDME] transactions is made in terms of functions and data attributes

### B.2.1 Comparison of functions

Parameter	PISTA (Level 1 / Level 2)	CARDME	Remarks
<b>Layer 7 services and EFC functions</b>			
INITIALISATION	Yes	Yes	
GET	Yes	Yes	
SET	Yes	Yes	
ACTION GET_STAMPED	No/Yes	Yes	Not used in the [PISTA] level 1 transaction, but it is a mandatory functionality also for "PISTA 1 OBU" in order to ensure seamless migration to level 2
ACTION SET_MMI	Yes	Yes	
ACTION ECHO	Yes	Yes	
EVENT-REPORT RELEASE	Yes	Yes	
<b>Security features</b>			
Transaction counter	No	Yes	RSE and central system functionality
Static authenticator	Yes, Receipt Vehicle Contract	Yes, Receipt	RSE / central system functionality.
Dynamic authenticator	No / Yes, Issuer	Yes, Issuer / Operator	RSE and OBU functionality, keys derived from PAN and ContractProvider (8 keys). Not used in [CESARE] / [PISTA] level 1 transaction, but a mandatory OBU feature as required for seamless migration to level 2
Data access protection	No	Yes, static / dynamic	RSE and OBU functionality. Data access keys derived from "OBUGroupId"

**Table B. 2** Comparison of PISTA and CARDME functions

The same sub-set of layer 7 services and EFC functions is used in all transactions except for [PISTA] level 1. However, it is mandatory that the "PISTA OBU" supports the GET\_STAMPED function from the start in order to ensure seamless migration to level 2.

It is important to notice that data access protection (i.e. access credentials) is used by [CARDME], whereas it is not used by [PISTA], in order to provide a means to

- Protect against non-authorized access to sensitive user data
- Protect against (commercial) use of the OBU by a non-authorized operator

It should also be noted that [PISTA] does not make use of the transaction counter. However, the transaction counter is a function residing in the RSE and central system (and not in the OBU).

<sup>14</sup> The PISTA and CARDME transactions are accounted for in 5.1.

## B.2.2 Comparison of information security

Information security		PISTA (levels 1 / 2)	CARDME	Remarks
Contract (R)	Data integrity	Y	N	ContractAuthenticator provides data integrity
	data origin authentication	N / Y	N	
	data access protection	N	N	
Payment Means (R)	Data integrity	Y	Y	
	data origin authentication	N	Y	
	data access protection	N	Y	
Receipt (R/W)	Data integrity	Y	Y	
	data origin authentication	N	N	
	data access protection	N	Y	
Vehicle (R)	Data integrity	Y	N	VehicleAuthenticator provides data integrity
	data origin authentication	N	N	
	data access protection	N	Y	
Equipment (R – R/W)	Data integrity	N	N	
	data origin authentication	N	N	
	data access protection	N	Y	

**Table B. 3** Comparison of PISTA and CARDME information security

CARDME focuses the security on the payment means information. It also provides a means to prevent non-authorised (read and write) access to data.

## B.2.3 Comparison of data attributes

The table below provides a comparison of the data attributes in the OBU. What attributes that are used in a transaction is controlled by the RSE. The letters in the table below have the following meaning:

- VST: attribute always exchanged as part of the vehicle service table (VST);
- R: attribute read during an EFC transaction;
- W: attribute written during an EFC transaction.
- X: available
- --: not available

ATTRIBUTES (EID>0)		PISTA (levels 1/2)	CARDME	Remarks
Name	AttrId			
CONTRACT				Information associated with the service rights of the issuer of the EFC service
EFCCContextMark (VST)	0	X	X	
ContractAuthenticator (R)	4	X	--	5 (=1+4) octets long in [PISTA]
PAYMENT				Data associated with the payment transaction
Payment Means (R)	32	X	X	
VEHICLE				Information pertaining to the identification and characteristics of the vehicle
VehicleLicencePlateNumber (R)	16	--	X (optional)	
VehicleClass (R)	17	X	X (optional)	
VehicleDimensions (R)	18	X	X (optional)	
VehicleAxles (R)	19	X	X (optional)	
VehicleWeightLimits (R)	20	--	X (optional)	
VehicleSpecificCharacteristics (R)	22	--	X (optional)	
VehicleAuthenticator (R)	23	X	--	5 (=1+4) octets long in [PISTA]
RECEIPT				Information associated with a specific session, including both financial and operational data.
ReceiptData1 (R/W)	33	X	X	
ReceiptData2 (R/W)	34	X	X	
ReceiptText (W / display)	12	--	X	
EQUIPMENT				
EquipmentOBUId (R)	24	X	--	
EquipmentStatus (R/W)	26	X	X	

**Table B. 4** Comparison of PISTA and CARDME application data

It should be noted that PISTA does not include Vehicle Specific Characteristics (including Environmental Characteristics, Engine Characteristics etc.) and Vehicle Class (including vehicle trailer indication) that are often desired / required for HGV charging schemes. PISTA and this specification include Contract Authenticator and Vehicle Authenticator.

CARDME provides support for pre-configured and personalised OBUs [CARDME, Part 2, 3.2.2]. The former does not contain Vehicle Licence Plate Number, Vehicle Dimensions, Vehicle Axles, Vehicle Weight Limits, Vehicle Specific Characteristics (but contains Vehicle Class) and is typically envisaged for passenger cars. The latter contains all vehicle data and is typically envisaged for HGV.

Note also that CARDME provides the possibility to display the receipt via the OBU's MMI (e.g. a display).

## ANNEX C - CLASSIFICATION CONCEPTS

This annex briefly discusses the different classification concepts enabled by this EFC-DSRC specification. It should be noted that adoption of a specific classification scheme is not within the scope of this specification, but should be handled in a MoU or a local system set-up. This chapter explains the three concepts for classification that is enabled by the specification.

In most systems the fee charged is dependent on the “class” the vehicle belongs to. Determining the class is generally known as “classification” of the vehicle. There are two basic methods for this:

- **Measured** classification is commonly used at motorway tolling with mono-lane toll stations and where manual payment is common. This means that the system measures the relevant vehicle characteristics and determines the vehicle class (it can also simply be the toll attendant looking at the vehicle).
- **Declared** classification is preferred for urban tolling and HGV-charging schemes. It might multi-lane tolling stations or a more complex set of vehicle characteristics that are needed and that may be difficult to measure (e.g. weight, or engine characteristics). In declared classification the RSE reads classification data stored in the OBU and determines the class from those data. This requires storage of correct data in the OBU, and checks towards fraud with those data (or the use of the wrong OBU in a vehicle).

We can analyse the needs of three different types of operators.

- **Bridge and motorway tolls.** These are typically mono-lane and often use measured classification. They have no need to provide clients with more advanced OBUs.
- **Urban tolling.** In this case the environment is often multi-lane (no space for toll plazas). In some cases charging schemes may also include discounts etc. for special vehicles (like environmental friendly cars). Declared classification is needed, but most vehicles are normal passenger cars, and for them there is no need for a large set of vehicle data. OBUs for urban tolling are often very price sensitive to the operator / users.
- **HGV-charging schemes.** These systems may need to perform detailed calculations of the fee depending on a large range of parameters. Declared classification is needed with several vehicle characteristics accessed. OBUs are often advanced and expensive.

As both classification methods (measured and declared) are relevant for different situations and different operators, this specification does not rule out the use of any method, but enables both to be used in parallel. This specification allows for three different sub-contracts for the user when acquiring the OBU, depending on the vehicle data stored on the OBU. The EFC transaction is the same for all sub-contracts, but the amount of data retrieved during the transaction differs. This specification offers three distinct sets of vehicle data as below.

Vehicle data	Sub-contract		
	Type 1 Full vehicle data	Type 2 Pre-defined class	Type 3 No vehicle data
VehicleClass	Entry	Entry	Dummy
VehicleLicencePlateNumber	Entry	Dummy	Dummy
VehicleDimensions	Entry	Dummy	Dummy
VehicleAxles	Entry	Dummy	Dummy
VehicleWeightLimits	Entry	Dummy	Dummy
VehicleSpecificCharacteristics	Entry	Dummy	Dummy

**Table C. 1** EFC sub-contracts and associated sets of vehicle data

**Type 1 - Full vehicle data** means that all the above listed vehicle data is stored in the OBU. This OBU may not be moved between vehicles. Vehicle data must be correct and may require regular updating.

**Type 2 - Pre-defined class** means that a pre-defined vehicle class<sup>15</sup> is stored in the OBU. This type of OBU may be moved between vehicles of the same class (e.g. normal passenger cars).

**Type 3 - No vehicle data** means that no vehicle data at all is stored in the OBU. Hence, the OBU may freely be moved between vehicles. The type is allowed during a migration period to enable local operators (with few pan-European clients) to use the specification from the start.

Enabling three different sub-contracts for classification raises the question if these can work together. What happens if an OBU with one type of contract is used in a system that normally would require another type? The different situations are summarised in the table below.

Operator	Sub-contract		
	Type 1 Full vehicle data	Type 2 Pre-defined class	Type 3 No vehicle data
Bridge and motorway tolls	Measured: OK	Measured: OK	Measured: OK
Urban tolling	Declared: OK	Declared: OK	Risk for high fee (1)
HGV-charging	Declared: OK	N/A <sup>16</sup>	Not OK (2)

**Table C. 2** Typical handling of the three sub-contracts by different type of operators

As can be seen most cases present no problem for classification, in fact there are only two special cases that may present trouble. These two are numbered in the table above and discussed below:

1. In this case there is a risk for a too high fee for the user (the user gets the maximum fee as no classification data is presented by the OBU). However, the highest fee will probably be quite low in urban tolling. This is a risk that the user will take by using an intermediate classification solution (type 3). For full interoperable use the user should instead acquire another OBU (type 1 or 2) or use an alternative payment method (e.g. manual, video registration).
2. In this case there is also a risk for a too high fee for the user (the same as above). However, there are two exceptions; the fee might be quite high and HGV-charging systems might want to use a wide range of vehicle characteristics for the fee calculation. In this case the use of type 3 sub-contract is not recommended, and the user would have to acquire another OBU (type 1) for interoperable use or use an alternative payment method if offered by the operator.

The conclusion is that this three-piece classification solution can work fine in almost all cases, as well as being in line with user and operator requirements for both short and long term. It must, however, be emphasised that this specification only enables such a solution, it still needs to be clarified in terms of procedures and contracts in a MoU between the operators.

<sup>15</sup> This class is defined for Swedish use in this specification. A common European set of vehicle classes may be defined in European co-operation.

<sup>16</sup> As there are only cars using type 2 sub-contracts.

## ANNEX D - DATA SPECIFICATION

Annex D specifies the EFC application data.

Attribute Id / Name Data element	Definition & remarks	Usage	Length In octets
<b>CONTRACT</b>	Information associated with the service rights of the issuer of the EFC service.		
<b>0 / EFC-ContextMark</b>			<b>6</b>
ContractProvider	Identifies the organisation that issued the service rights given in the Contract and has an overall responsibility of all data (listed in this specification) stored in the OBU. Numbers shall be assigned on a national basis. The ContractProvider might, e.g., be a single operator or a group of operators holding an inter-company agreement.	Usage according to [CARDME]: Country code (10 bits) + Issuer Identifier (14 bits). Example : <ul style="list-style-type: none"> <li>Sweden= 10100 10000'B</li> <li>Issuer = XX XXXX XXXX XXXX'B, e.g. <ul style="list-style-type: none"> <li>00 0000 0000 0001'B = Öresundskonsortiet</li> <li>00 0000 0000 0011'B = EuroPark Svenska AB</li> <li>00 0000 0000 0101'B = Scandlines AB</li> </ul> </li> </ul> See also ENV ISO 14816 Register at <a href="http://www.nen.nl/cen278">www.nen.nl/cen278</a> .	3
TypeOfContract	ContractProvider-specific designation of the rules that apply to the Contract. Allows, e.g., for the determination of the tariff or designating the type of purse associated with the contract.	A two octet value identifying the EFC contract residing in the OBU (e.g. central account or on-board purse). MMMM MMMM SSSS SSSS, where MMMM MMMM identifies the main type of contract (coding to be defined in MoU), and SSSS SSSS the sub-types. <ul style="list-style-type: none"> <li>SSSS SSSS = '00'H : not further specified</li> <li>SSSS SSSS = '01'H : Full vehicle data</li> <li>SSSS SSSS = '02'H : Pre-defined class</li> <li>SSSS SSSS = '03'H : No vehicle data</li> </ul>	2
ContextVersion	ContextVersion denotes the implementation version of the concerned contract within the context of the given ContractProvider, value assigned at the discretion of the ContractProvider. The ContextVersion is also used to identify the associated set of master keys.	Identification of the version residing in the OBU, also used as a security key reference for OBU issued according to "Basic Requirements Specification for Interoperable EFC-DSRC Systems in Sweden". 0sss vvvv'B, where <ul style="list-style-type: none"> <li>sss identifies the set of master keys used. 000'B identifies Security Key version 0.</li> <li>vvvv identifies the version of "Basic Requirements Specification for Interoperable EFC-DSRC Systems in Sweden". 0000'B = version 1.0</li> </ul>	1
<b>4 / ContractAuthenticator</b> ContractAuthenticator	Authenticator calculated by the ContractProvider when issuing the Contract, to prevent undetected tampering with contract data.	Usage according to [PISTA]. ContractAuthenticator carries the result of a cryptographic calculation, done by the issuer and unknown to the operator, using the EFCContextMark and PaymentMeans. Each issuer must define a cryptographic algorithm to calculate the authenticator, compulsory to initialise this authenticator by the issuer. Used for post/processing checking of the information integrity by the issuer.	<b>5 (1+4)</b>

**Table D.1** Data specification – EFC-ContextMark and ContractAuthenticator

Attribute Id / Name Data element	Definition & remarks	Usage	Length In octets
<b>PAYMENT</b>	Data associated with the payment transaction		
<b>32 / PaymentMeans</b>			<b>14</b>
Personal Account Number	Coded according to financial institutions.	PersonalAccountNumber ::= OCTET STRING (SIZE(10)) <ul style="list-style-type: none"> <li>• issuer identification number (IIN, 6 BCD), identifies the issuers of the PAN. The individual account number shall be assigned by the card issuing institution. The first digit is the major industry identifier.</li> <li>• Major Industry Identifier (1 BCD)               <ul style="list-style-type: none"> <li>- 0: for assignment by ISO/TC 68 and for other future industry assignments</li> <li>- 1: airlines</li> <li>- 2: airlines and other future industry assignments</li> <li>- 3: travel and entertainment</li> <li>- 4: banking/financial</li> <li>- 5: banking/financial</li> <li>- 6: merchandizing and banking</li> <li>- 7: petroleum</li> <li>- 8: health care, telecommunications and other future industry assignments</li> <li>- 9: for assignment by national standards bodies</li> </ul> </li> <li>• individual account number (variable length, max 12 digits, see ISO 7811-3), identifies an account in the CS.</li> <li>• check digit, providing a means to check that the PAN has not been modified. The check digit shall be calculated on preceding digits and be computed according to the Luhn formula for modules 10 check digit (see Annex B in ISO 7812-1).</li> <li>• padding bits set to '1'B, in order to accomplish a total length of 10 octets (including the check digit)</li> </ul>	10
PaymentMeans ExpiryDate	Expiring date of payment means. Payment means expires at 24h of PaymentMeans ExpiryDate	DateCompact	2
PaymentMeans UsageControl	issuer's specified restrictions on the geographic usage and services allowed for the applications	OCTET STRING (SIZE(2))::=00 00'H : no specified restrictions	2

**Table D.2** Data specification – PaymentMeans

Attribute Id / Name Data element	Definition & remarks	Usage	Length In octets
<b>VEHICLE</b>	Information pertaining to the identification and characteristics of the vehicle		
<b>16 / Vehicle License Plate Number</b> VehicleLicence PlateNumber	Claimed licence plate of the vehicle	Usage according to [CARDME]. Claimed licence plate of the vehicle, the length of the padded LPN is fixed to 10 octets. An LPN which is shorter than 10 characters is padded with NUL characters so as to achieve a total of 10 characters. Example : SE, LatinAlphabethNo1, OCD560 <ul style="list-style-type: none"> <li>Country code = SE = 1010010000'B</li> <li>Alphabet indicator = LatinAlphabethNo1 = 000000'B</li> <li>Length determinant = 10 octets = 00001010'B</li> </ul> LPN = OCD560 = 4F 43 44 35 36 30 00 00 00 00'H	<b>13</b>
<b>17/ Vehicle Class</b> Vehicle Class	Service provider specific information pertaining to the vehicle.	Vehicle class' substructure TCCC LLLL, where <ul style="list-style-type: none"> <li>T (trailer indicator) : <ul style="list-style-type: none"> <li>0'B = no trailer, also used the default value</li> <li>1'B = trailer present</li> </ul> </li> <li>CCC (Europea MoU) =000'B</li> <li>LLLL (Swedish vehicle classes): value assignments according to the below: <ul style="list-style-type: none"> <li>0000'B = not further defined</li> <li>0001'B = motorcycle</li> <li>0010'B = small passenger car (e.g. VW Lupo)</li> <li>0011'B = medium passenger car (e.g. Volvo S40)</li> <li>0100'B = big passenger car (e.g. Renault Espace)</li> <li>0101'B = bus with 2 vehicle axles</li> <li>0110'B = bus with 3 vehicles axles</li> <li>0111'B = Heavy Vehicle with 2 axles</li> <li>1000'B = Heavy Vehicle with 3 axles – Class 2</li> <li>1001'B = Heavy Vehicle with 4 axles – Class 3</li> <li>1010'B = vehicles exempt from road user charging fee</li> <li>1011'B = reserved for future assignment</li> <li>11LL'B = reserved for future assignments</li> </ul> </li> </ul>	<b>1</b>
<b>18/ Vehicle Dimensions</b>	Nominal vehicle dimensions according to [ISO 612].	Usage according to [EFC AID].	<b>3</b>
VehicleLengthOverall	Nominal maximum overall length of the vehicle according to [ISO 612], in dm, rounded to the next dm.	Example: a 6.15 m long vehicle is coded as 0011 1101'B	<b>1</b>
VehicleHeightOverall	Nominal overall unladen height, according to [ISO 612], in dm, rounded to the next dm.	Example: a 2.43 m high vehicle is coded as 0001 1000'B	<b>1</b>
VehicleWidthOverall	Nominal overall width, according to [ISO 612], in dm, rounded to the next dm	Example: a 1.87 m wide vehicle is coded as 00001 0010'B	<b>1</b>

**Table D.3** Data specification – Vehicle LicensePlateNumber and VehicleDimensions

Attribute Id / Name Data element	Definition & remarks	Usage	Length In octets
<b>19/ VehicleAxles</b>	Tyre type and number of axles, including drop axles. / Also gives information on the usage of dual tyres	Usage according to [EFC AID].	<b>2</b>
VehicleFirstAxleHeight	Bonnet height, measured over the front axle, in dm, rounded to the next dm.	Example: a bonnet height of 115 cm is coded as 0000 1011'B.	1
VehicleAxlesNumber	Tyre type and number of axles, including drop axles. / Also gives information on the usage of dual tyres	Tyre type and number of axles, including drop axles, encoded as follow: ttaa aaaa where <ul style="list-style-type: none"> <li>• tt : tyre type (2 bits) <ul style="list-style-type: none"> <li>• 00'B = not specified;</li> <li>• 01'B = single pair of tyres per axles</li> <li>• 10'B = dual pair of tyres per axles</li> <li>• 11'B = reserved for future CEN Use.</li> </ul> </li> <li>• aa aaaa: number of Axles (6 bits), including drop Axles. Example: 2 axles is encoded as 00 0010'B.</li> </ul>	1
<b>20/ Vehicle WeightLimits</b>	Vehicle weight limits according to [ISO 1176].	Usage according to [EFC AID].	<b>6</b>
VehicleMaxLadenWeight	Maximum permissible total weight including payload, according to [ISO 1176]. 10kg units, rounded down to the next 10kg step.	Example: a weight of 16'801 kg is coded as 0000 0110 1001 0000'B.	2
VehicleTrainMaximumWeight	Maximum permissible weight of the complete vehicle train, as defined in [ISO 1176]. 10kg units, rounded down to the next 10kg step. / ISO [1176] Code ISO-M18 maximum design mass of vehicle combination	Example: a weight of 16'801 kg is coded as 0000 0110 1001 0000'B.	2
VehicleWeightUnladen	Nominal unladen weight, according to [ISO 1176] in 10kg units, rounded down to the next 10kg step.	Example: a weight of 3'530 kg is coded as 0000 0001 0110 0001'B.	2

**Table D.4** Data specification – VehicleAxles and VehicleWeightLimits

Attribute Id / Name Data element	Definition & remarks	Usage	Length In octets
<b>22/ VehicleSpecific Characteristics</b> Vehicle Specific Characteristics	Further vehicle characteristics. Each enumerated value has a specific meaning assigned. The meaning of some values are defined in [EFC AID], others are reserved for future needs. Assignment of meaning to the unassigned enumerated values is subject to registration according to the registration procedure specified in EN 12834.	Usage according to [EFC AID]. <ul style="list-style-type: none"> <li>• Euro type (4 bits). As defined in EC directive 88/77/EEC annex 1, and in 91/542/EEC               <ul style="list-style-type: none"> <li>• 0000'B = no entry</li> <li>• 0001'B = euro 1</li> <li>• 0010'B = euro 2</li> <li>• 0011'B = euro 3</li> </ul> </li> <li>• Cop type (4 bits) : 0000'B = no entry</li> <li>• Engine characteristics (8 bits) :               <ul style="list-style-type: none"> <li>• 0000 0000'B = no entry</li> <li>• 0000 0001'B = no engine</li> <li>• 0000 0010'B = petrol unleaded</li> <li>• 0000 0011'B = petrol Leaded</li> <li>• 0000 0100'B = diesel</li> <li>• 0000 0101'B = LPG</li> <li>• 0000 0110'B = battery</li> <li>• 0000 0111'B = solar</li> </ul> </li> <li>• reservedForFutureCENUse (8-255)</li> <li>• Descriptive characteristics (8 bits) : as defined in prENV/278/8/1/5               <ul style="list-style-type: none"> <li>• 0000 0000'B = no entry</li> <li>• 0000 0001'B = vehicle shape 1. Etc.</li> </ul> </li> </ul>	4
<b>23/ VehicleAuthenticator</b> VehicleAuthenticator	Authenticator calculated by the entity entering the data elements at time of entry or modification.	Usage according to [PISTA]. VehicleAuthenticator: carries result of a cryptographic calculation, done by the issuer, using all the vehicle attributes. VehicleAuthenticator is calculated with a "common algorithm" created and distributed with the MoU, since all operators need to be able to check them. It is calculated by the issuer upon personalisation of the OBU with an algorithm known by the operators and the MoU. Each operator may decide whether to retrieve this attribute or not during the transaction, and therefore whether checking it or not, eventually in real time or in post processing.	5 = 1+4

**Table D.5** Data specification – VehicleSpecificCharacteristics and VehicleAuthenticator

Attribute Id / Name Data element	Definition & remarks	Usage	Length In octets
<b>EQUIPMENT</b>	Information pertaining to the OBU.		
<b>24 / EquipmentOBUID</b> EquipmentOBUID	Identification number of OBU. / The manufacturer ID is always exchanged as a part of the VST	The EquipmentOBUID shall be a unique identification number of the OBU. It is written into the OBU during the OBU manufacturing process.	<b>variable</b>
<b>26 / EquipmentStatus</b> EquipmentStatus	Operator-specific EFC application-related information pertaining to the status of the equipment. Boolean information to support an operator's handling of an OBU on application level. (E.g. 'next suitably equipped gantry should take an enforcement picture')	LLLL CCCC CCCC CCCC, where LLLL'B : Local use (4 bits), coding and use at the discretion of the operator. Shall be set to 0000'B : not specified. CCCC CCCC CCCC'B : sequential transaction counter (12 bits), shall be set to 0000 0000 0000'B : upon personalisation.	<b>2</b>
<b>RECEIPT</b>	Information associated with a specific session, including both financial and operational data.		
<b>33 / ReceiptData1</b>	Latest receipt.		<b>28</b>
SessionTime	Date and Time of session with a two-seconds resolution. Time Easy to decode into a displayable format by OBU. Date and time value assignment – Octet Aligned[01.01.1990, 00:00:00]... [31.12.2116, 21:59:58], then rollover.	Usage according to [CARDME, Annex 3] : Example : 1st of March 2003, 21:12:10 is encoded as year (1990..2117); 000 1101'B month (0..12); 0011'B date (0..31); 00001'B hours (0..23); 10101'B minutes (0..59); 001100'B double-secs, 2 s resolution (0..29); 00101'B	4
SessionServiceProvider	Operator that provides the service of the session. Provider Identifier of an operator.	See ContractProvider in the EFC-ContextMark	3
StationLocation	Service provider specific coding of the station location. Toll plaza code defined by country organisation.	Usage according to [CARDME].	2
SessionLocation	INT1 Travel direction + Lane Code0/1 + 0..127	Usage according to [CARDME].	1
SessionType	Designates the type of service station.	Usage according to [CARDME].	1
SessionResult	Code designating whether a session has been completed successfully or not.	Usage according to [CARDME].	1
SessionTariffClass	Service provider specific tariff class applied in the session. Enables to reproduce the price calculation (e.g. claimed or measured vehicle class that was applied.)	Usage according to [CARDME].	1
SessionClaimedClass	Service provider specific vehicle class derived from claimed characteristics in the data group Vehicle. Claimed class and applied class (tariff class) may differ.	See Vehicle Class.	1
<b>34 / ReceiptData2</b>	Penultimate receipt	See ReceiptData1	<b>28</b>

**Table D.6** Data specification – EquipmentOBUID, EquipmentStatus, ReceiptData1 and ReceiptData2

Data element	Definition & remarks	Usage	Length In octets
AuthenticationKey1	Private	A key used to compute authenticators (see 6.3). KeyRef = 111	8
AuthenticationKey2	Private	Idem KeyRef = 112	8
AuthenticationKey3	Private	Idem KeyRef = 113	8
AuthenticationKey4	Private	Idem KeyRef = 114	8
AuthenticationKey5	Private	Idem KeyRef = 115	8
AuthenticationKey6	Private	Idem KeyRef = 116	8
AuthenticationKey7	Private	Idem KeyRef = 117	8
AuthenticationKey8	Private	Idem KeyRef = 118	8
AccessKeys	Private. AccessKeys are in CARDME but not in PISTA transactions.	The access key is the key used to compute Access credentials (see 6.3).	8
AC_CR	Access credentials calculated by the RSE and the OBU using RndOBU and the Access Key AC_CRKey.	Integer (0..4'294'967'295). <ul style="list-style-type: none"> <li>AC_CR = PW = 04 94 F8 97'H. Compute AC_CR(0) according to the DES algorithm [DEA], see also 6.3.</li> <li>AC_CR = DES. Compute AC_CR(k) according to the DES algorithm [DEA], see also 6.3.</li> </ul>	1+4
AC_CR-KeyReference	Reference to the key generation and the Diversifier for the computation of AC_CRKey.	<ul style="list-style-type: none"> <li>Key reference (k): Integer (0..255) (8 bits)</li> <li>Diversifier: Integer (0..255). (8 bits)</li> </ul> Example : Key reference (# 1) and Diversifier # 2 : <ul style="list-style-type: none"> <li>0000 0001'B (Key reference (1)):</li> <li>0000 0010'B (Diversifier(2)).</li> </ul>	2
KeyRef	Reference to AuKey used for the computation of the Authenticators, e.g. Issuer and Operator Authenticators. The Issuer decides which keys are shared with (MoU) Operators (referenced through AuKey_Op), and which are only known by himself (referenced through AuKey_Iss).	Integer (0..255). Example: AuthenticationKey1 reference (=111 <sub>10</sub> )	1
RndOBU	Random number (nonce) used together with AccessKey (referenced through AC_CR-KeyReference) to calculate the Access credentials.	Usage according to [CARDME]. Integer (0..4'294'967'295)	5 = 1+4
RndRSE	Random number, containing SessionTime, from RSE used for the computation of Authenticator.	Usage according to [CARDME]. ReceiptData1.SessionTime preceded by a length determinant.	5 = 1+4

**Table D.7** Data specification – Authentication keys, access keys and arguments in security related EFC functions.

## ANNEX E - SECURITY IMPLEMENTATION EXAMPLES

Annex D illustrates the defined cryptographic mechanisms by means of a few numerical examples. The **ede[Key] (VAL)** syntax is used below for description of the TripleDES operations, where

- 'ede' denotes encryption, decryption and encryption according to the TripleDES algorithm;
- '[Key]' denotes the applied key;
- '(VAL)' denotes the input value to the TripleDES operations.

Numeric values in the examples below are in hexadecimal.

### E.1 Key derivations

The OBU specific keys<sup>17</sup> are derived from the Master keys by TripleDES.

#### E.1.1 Authentication keys

Below is described the procedure for derivation of the Authentication keys, applicable for both the PISTA and CARDME transactions.

The OBU-specific keys are derived from the Masterkeys by TripleDES. 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 converting the 64 bits PM\_8 to 32 bits with the following algorithm:  
$$\text{Compact\_PaymentMeans} = [\text{HighDWord32}(\text{PM\_8})] \text{ XOR } [\text{LowDWord32}(\text{PM\_8})]$$
3. Pad left the concatenation of Compact\_PaymentMeans || ContractProvider with '00' to obtain an 8 octets value VAL:  
$$\text{VAL} = \text{'Compact\_PaymentMeans || ContractProvider || 00'}$$
4. Compute the AuthenticationKey(i) as follows:  
$$\text{AuthenticationKey}(i) = \text{ede}[\text{MAuKey}(i)] (\text{VAL})$$

In the example below we use the following application data and Master Key values:

- PaymentMeans:
  - PersonalAccountNumber (PAN): '52 75 12 34 56 78 90 12 FF FF' (16 characters PAN, padding with '1' bits)
  - PaymentMeansExpiryDate: '21 21' (2006-09-01)
  - PaymentMeansUsageControl: '00 00'
- ContractProvider: 'A4 00 01' (Sweden (SE), Issuer #1 (Öresundskonsortiet))
- MAuKey : ,13 13 13 13 13 13 13 13 AB AB AB AB AB AB AB AB';

The CompactPAN is calculated as:

- CompactPAN = Sub(PAN, 0, 4) xor Sub(PAN, 4, 4) = '04 0D 82 26'
- {Sub(PAN,0,4) = HighDWord32(PAN), Sub(PAN, 4,4) is LowDWord32(PAN)}

---

<sup>17</sup> The key management is outside the scope of this specification. It is assumed that the key management foresees two set of keys ("contract issuer keys" and "Operator keys"), for calculations of Issuer and Operator authenticators.

The input data (VAL) follow from:

- VAL = 'CompactPAN || ContractProvider || 00' = '04 0D 82 26 A4 00 01 00'

With [DEA] defining the Initial Chaining Value ICV:

- ICV : '00 00 00 00 00 00 00 00',

this gives the following value for the Authenticator Key:

- AuthenticationKey = ede[MAUKey](VAL) = '26 BF 3D F3 BC E3 65 6B'

where the formula above denotes the Triple-DES operation using MAUKey over the data string VAL.

With a different value for the PAN the derived key will be different. As an example, with a PAN = '58 61 12 34 56 78 90 12 FF FF', and the same Contract Provider and Master Keys as above we find the following derived keys:

- AuthenticationKey = 'A7 AD C3 82 44 1C 1D 00'

### E.1.2 Access credentials key

Below is described the derivation of the Access credentials key, that is used in the CARDME transaction.

Use the following procedure to compute the AC\_CRKey(k) for a given AC\_CR-MasterKeyReference. (k):

1. Set AC\_CR-Reference = 'AC\_CR-MasterKeyRef || AC\_CR-Diversifier' (2 octets)
2. Make the concatenation of 'AC\_CR-Reference || AC\_CR-Reference || AC\_CR-Reference || AC\_CR-Reference' to obtain an 8 octets value VAL:  
VAL = 'AC\_CR-Reference || AC\_CR-Reference || AC\_CR-Reference || AC\_CR-Reference'
3. Compute the AC\_CRKey(k) as follows:  
AC\_CRKey(k) = ede[MAC\_CRKey(k)] (VAL)

We use the following application data values and Master Key in the example below:

- AC\_CR-Reference:
  - AC\_CR-MasterKeyRef.: '12'
  - AC\_CR-Diversifier: '34'
- MAC\_CRKey : '57 57 57 57 57 57 57 57 EF EF EF EF EF EF EF EF'

This gives:

- VAL = 'AC\_CR-Reference || AC\_CR-Reference || AC\_CR-Reference || AC\_CR-Reference' = '12 34 12 34 12 34 12 34'

And:

- AC\_CRKey = ede[MAC\_CRKey](VAL) = '9B 48 AA E0 7A 7B C0 08'

Again, a different AC\_CR-Reference or Master Key will produce a completely different Access Credentials Key.

For calculation of the AC\_CRKey(0) to be used to calculate the fixed "password" in the initial phases of the security level implementations, the value MAC\_CRKey(0) = 0, AC\_CR-MasterKeyRef. = 0 and AC\_CR-Diversifier = 0 shall be used. This will give the fixed value of the AC\_CRKey(0) = '8C A6 4D E9 C1 B1 23 A7'.

## E.2 Computation of authenticators

The computation of authenticators is performed according to [DEA].

### E.2.1 ContractAuthenticator authenticators

This Authenticator is used in the PISTA transaction.

We use the following values:

- Contract Authenticator = '12 34 56 78'
- RndRSE: '1A 61 A9 85' (1<sup>st</sup> of March 2003, 21:12:10)
- AuKey = '26 BF 3D F3 BC E3 65 6B'

The message M (the input data) is then equal to:

- M = 'AttributeList (including ContractAuthenticator) || RndRSE (octet string) || Padding' = '01 04 24 04 12 34 56 78 04 1A 61 A9 85 00 00 00'

and

- D1 = I1 = Sub(M, 0, 8) = '01 04 24 04 12 34 56 78'
- D2 = Sub(M, 8, 8) = '04 1A 61 A9 85 00 00 00'

With ICV : '00 00 00 00 00 00 00 00': the Input I1 = ICV xor D1 = '01 04 24 04 12 34 56 78'  
O1 = e[AuKey]( I1 ) = 'F4 E8 03 87 7F FE E4 2E'

and

I2 = O1 xor D2 = 'F4 E8 03 87 7F FE E4 2E' xor '04 1A 61 A9 85 00 00 00' = 'F0 F2 62 2E FA FE E4 2E'

Calculation of O2 gives:

O2 = e[AuKey]( I2 ) = '98 2D B2 22 3D AD D5 14'

The leftmost 32 bits represent the Authenticator:

- Auth = Sub(O2, 0, 4) = '98 2D B2 22'

A change in the input parameters will completely change the Authenticators. To illustrate this we calculate the Authenticator for a different value of RndRSE, without changing the values for the other parameters.

With:

- RndRSE: '1A 61 A9 86'

we find:

- Auth = '9B 3C 8C 94'.

## E.2.2 Payment means authenticators

We use the following values:

- PaymentMeans = '52 75 12 34 56 78 90 12 FF FF 21 21 00 00'
- RndRSE: '1A 61 A9 85' (1<sup>st</sup> of March 2003, 21:12:10)
- AuKey\_Iss = '26 BF 3D F3 BC E3 65 6B'

The message M (the input data) is then equal to:

- M = 'AttributeList (including PaymentMeans) || RndRSE (octet string) || Padding' = '01 20 40 52 75 12 34 56 78 90 12 FF FF 21 21 00 00 04 1A 61 A9 85 00 00'

and

- D1 = I1 = Sub(M, 0, 8) = '01 20 40 52 75 12 34 56'
- D2 = Sub(M, 8, 8) = '78 90 12 FF FF 21 21 00'
- D3 = Sub(M, 16, 8) = '00 04 1A 61 A9 85 00 00'

With ICV : '00 00 00 00 00 00 00 00': the Input I1 = ICV xor D1 = '01 20 40 52 75 12 34 56'

O1 = e[AuKey\_Iss]( I1 ) = '45 1F F4 A9 72 9B CE 58'

and

I2 = O1 xor D2 = '45 1F F4 A9 72 9B CE 58' xor '78 90 12 FF FF 21 21 00' = '3D 8F E6 56 8D BA EF 58'

Calculation of O2 gives:

- O2 = e[AuKey\_Iss]( I2 ) = '4F BB E8 C6 4B 0B EF A5'

and

- I3 = O2 xor D3 = '4F BB E8 C6 4B 0B EF A5' xor '00 04 1A 61 A9 85 00 00' = '4F BF F2 A7 E2 8E EF A5'

Calculation of O3 gives:

- O3 = e[AuKey\_Iss]( I3 ) = 'BF 87 5F F0 90 AD BF E0'

The leftmost 32 bits represent the Issuer Authenticator:

- Auth = Sub(O3, 0, 4) = 'BF 87 5F F0'

A change in the input parameters will completely change the Authenticators. To illustrate this we calculate the Authenticator for a different value of RndRSE, without changing the values for the other parameters.

With:

- RndRSE: '1A 61 A9 86' (1<sup>st</sup> of March 2003, 21:12:12)

we find:

- Auth = '23 98 AA 8D'.

## E.3 Computation of Access Credentials

The Access credentials are used in the CARDME transaction.

We use the AC\_CRKey = '9B 48 AA E0 7A 7B C0 08' derived in E.1.2.

Assuming

- RndOBU = '97 86 75 64' gives
- VAL = 'RndOBU || 00 00 00 00' = '97 86 75 64 00 00 00 00'.

Calculation gives:

- O1 = e[AC\_CRKey](VAL) = 'E0 55 EA 12 1F 5C 97 D7' and hence:
- AC\_CR = Sub(O1, 0, 4) = 'E0 55 EA 12'

A fixed AC\_CR value is used for AC\_CRMasterKeyRef = k = 0). This value may also be calculated by first deriving the AC\_CRKey from the MAC\_CRKey and the AC\_CR-Reference as shown below:

- MAC\_CRKey(0) = '00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00' and
- VAL = 'AC\_CR-Reference || AC\_CR-Reference || AC\_CR-Reference || AC\_CR-Reference' = '00 00 00 00 00 00 00 00'

this gives:

- AC\_CRKey = ede[MAC\_CRKey](VAL) = '8C A6 4D E9 C1 B1 23 A7'

Then the Access Credentials can be calculated with RndOBU = '00 00 00 00' which gives:

- VAL = 'RndOBU || 00 00 00 00' = '00 00 00 00 00 00 00 00' and
- O1 = e[AC\_CRKey](VAL) = '04 94 F8 97 08 87 1D 3F' and hence:
- AC\_CR(0) = '04 94 F8 97'