



# Economic and Social Council

Distr.: General  
17 January 2018  
English  
Original: English, French and  
Russian

---

## Economic Commission for Europe

Inland Transport Committee

Working Party on Road Transport

Group of Experts on European Agreement Concerning Work of  
Crews of Vehicles Engaged in International Road Transport (AETR)

Seventeenth session

Geneva, 19 February 2018

Item 2 (b) of the provisional agenda

Programme of Work

## Appendix 1C

Submitted by Estonia\*

This document contains Annex 1C to Regulation (EU) 2016/799. The document should be used together with amendment proposals included in ECE/TRANS/SC.1/GE.21/2018/1.


---

\* The document is reproduced as submitted.

GE.18-00678(E)



\* 1 8 0 0 6 7 8 \*

Please recycle 



<b>INTRODUCTION</b>	<b>9</b>
<b>1 DEFINITIONS</b>	<b>11</b>
<b>2 GENERAL CHARACTERISTICS AND FUNCTIONS OF THE RECORDING EQUIPMENT</b>	<b>18</b>
<b>2.1 General characteristics</b>	<b>18</b>
<b>2.2 Functions</b>	<b>19</b>
<b>2.3 Modes of operation</b>	<b>20</b>
<b>2.4 Security</b>	<b>22</b>
<b>3 CONSTRUCTION AND FUNCTIONAL REQUIREMENTS FOR RECORDING EQUIPMENT</b>	<b>22</b>
<b>3.1 Monitoring cards insertion and withdrawal</b>	<b>22</b>
<b>3.2 Speed, position and distance measurement</b>	<b>23</b>
3.2.1 .. Measurement of distance travelled	23
3.2.2 .. Measurement of speed	24
3.2.3 .. Measurement of position	24
<b>3.3 Time measurement</b>	<b>24</b>
<b>3.4 Monitoring driver activities</b>	<b>25</b>
<b>3.5 Monitoring driving status</b>	<b>25</b>
<b>3.6 Drivers entries</b>	<b>26</b>
3.6.1 .. Entry of places where daily work periods begin and/or end	26
3.6.2 .. Manual entry of driver activities and driver consent for ITS interface	26
3.6.3 .. Entry of specific conditions	28
<b>3.7 Company locks management</b>	<b>28</b>
<b>3.8 Monitoring control activities</b>	<b>29</b>
<b>3.9 Detection of events and/or faults</b>	<b>29</b>
3.9.1 .. “Insertion of a non-valid card” event	29
3.9.2 .. “Card conflict” event	29
3.9.3 .. “Time overlap” event	30
3.9.4 .. “Driving without an appropriate card” event	30
3.9.5 .. “Card insertion while driving” event	30
3.9.6 .. “Last card session not correctly closed” event	30
3.9.7 .. “Over speeding” event	30
3.9.8 .. “Power supply interruption” event	31

3.9.12 “Motion data error” event _____	31
3.9.14 “Security breach attempt” event _____	31
3.9.16 “Card” fault _____	32
3.9.17 “Recording equipment” fault _____	32
<b>3.10 Built-in and self-tests _____</b>	<b>32</b>
<b>3.11 Reading from data memory _____</b>	<b>33</b>
<b>3.12 Recording and storing in the data memory _____</b>	<b>33</b>
3.12.1. Equipment identification data _____	33
3.12.1.1 Vehicle Unit identification data _____	33
3.12.1.2 Motion sensor identification data _____	34
3.12.1.3 Global Navigation Satellite Systems identification data _____	35
3.12.2. Keys and Certificates _____	35
3.12.3. Driver or workshop card insertion and withdrawal data _____	35
3.12.4. Driver activity data _____	36
3.12.5. Places and positions where daily work periods begin, end, and/or where 3 hours continuous driving time is reached _____	36
3.12.6. Odometer data _____	37
3.12.7. Detailed speed data _____	37
3.12.8. Events data _____	37
3.12.9. Faults data _____	42
3.12.10 Calibration data _____	42
3.12.11 Time adjustment data _____	43
3.12.12 Control activity data _____	43
3.12.13 Company locks data _____	44
3.12.14 Download activity data _____	44
3.12.15 Specific conditions data _____	44
3.12.16 Tachograph card data _____	44
<b>3.13 Reading from tachograph cards _____</b>	<b>45</b>
<b>3.14 Recording and storing on tachograph cards _____</b>	<b>45</b>
3.14.1. Recording and storing in first generation tachograph cards _____	45
3.14.2. Recording and storing in second generation tachograph cards _____	46
<b>3.15 Displaying _____</b>	<b>46</b>
3.15.1. Default display _____	47

---

3.15.2. Warning display	48
3.15.3. Menu access	48
3.15.4. Other displays	48
<b>3.16 Printing</b>	<b>49</b>
<b>3.17 Warnings</b>	<b>50</b>
<b>3.18 Data downloading to external media</b>	<b>51</b>
<b>3.19 Remote communication for targeted roadside checks</b>	<b>51</b>
<b>3.20 Output data to additional external devices</b>	<b>52</b>
<b>3.21 Calibration</b>	<b>53</b>
<b>3.22 Roadside calibration checking</b>	<b>54</b>
<b>3.23 Time adjustment</b>	<b>54</b>
<b>3.24 Performance characteristics</b>	<b>54</b>
<b>3.25 Materials</b>	<b>55</b>
<b>3.26 Markings</b>	<b>55</b>
<b>4 CONSTRUCTION AND FUNCTIONAL REQUIREMENTS FOR TACHOGRAPH CARDS</b>	<b>56</b>
<b>4.1 Visible data</b>	<b>56</b>
<b>4.2 Security</b>	<b>60</b>
<b>4.3 Standards</b>	<b>60</b>
<b>4.4 Environmental and electrical specifications</b>	<b>60</b>
<b>4.5 Data storage</b>	<b>61</b>
4.5.1 .. Elementary files for identification and card management	61
4.5.2 .. IC card identification	61
4.5.2.1 Chip identification	62
4.5.2.2 DIR (only present in second generation tachograph cards)	62
4.5.2.3 ATR information (conditional, only present in second generation tachograph cards)	62
4.5.2.4 Extended length information (conditional, only present in second generation tachograph cards)	62
4.5.3 .. Driver card	62
4.5.3.1 Tachograph application (accessible to first and second generation vehicle units)	62
4.5.3.1.1 Application identification	62
4.5.3.1.2 Key and Certificates	62
4.5.3.1.3 Card identification	62
4.5.3.1.4 Card holder identification	63

4.5.3.1.5	Card download.....	63
4.5.3.1.6	Driving licence information.....	63
4.5.3.1.7	Events data.....	63
4.5.3.1.8	Faults data.....	64
For the purpose of this subparagraph, time shall be recorded with a resolution of 1 second. ....		64
4.5.3.1.9	Driver activity data .....	64
4.5.3.1.10	Vehicles used data .....	65
4.5.3.1.11	Places where daily work periods start and/or end.....	65
4.5.3.1.12	Card session data .....	66
4.5.3.1.13	Control activity data .....	66
4.5.3.1.14	Specific conditions data.....	66
4.5.3.2	Tachograph generation 2 application (not accessible to first generation vehicle unit) .....	66
4.5.3.2.1	Application identification .....	66
4.5.3.2.2	Keys and Certificates.....	66
4.5.3.2.3	Card identification .....	67
4.5.3.2.4	Card holder identification .....	67
4.5.3.2.5	Card download.....	67
4.5.3.2.6	Driving licence information.....	67
4.5.3.2.7	Events data.....	67
4.5.3.2.8	Faults data.....	68
4.5.3.2.9	Driver activity data .....	69
4.5.3.2.10	Vehicles used data .....	69
4.5.3.2.11	Places and positions where daily work periods start and/or end.....	70
4.5.3.2.12	Card session data .....	70
4.5.3.2.13	Control activity data .....	70
4.5.3.2.14	Specific conditions data.....	70
4.5.3.2.15	Vehicle units used data .....	71
4.5.3.2.16	Three hours continuous driving places data.....	71
4.5.4	.. Workshop card .....	71
4.5.4.1	Tachograph application (accessible to first and second generation vehicle units) .....	71
4.5.4.1.1	Application identification .....	71
4.5.4.1.2	Keys and Certificates.....	71
4.5.4.1.3	Card identification .....	72
4.5.4.1.4	Card holder identification .....	72

---

4.5.4.1.5	Card download.....	72
4.5.4.1.6	Calibration and time adjustment data .....	72
4.5.4.1.7	Events and faults data .....	72
4.5.4.1.8	Driver activity data .....	73
4.5.4.1.9	Vehicles used data .....	73
4.5.4.1.10	Daily work periods start and/or end data .....	73
4.5.4.1.11	Card session data .....	73
4.5.4.1.12	Control activity data .....	73
4.5.4.1.13	Specific conditions data .....	73
4.5.4.2	Tachograph Generation 2 application (not accessible to first generation vehicle unit).....	73
4.5.4.2.1	Application identification .....	73
4.5.4.2.2	Keys and Certificates.....	73
4.5.4.2.3	Card identification .....	74
4.5.4.2.4	Card holder identification .....	74
4.5.4.2.5	Card download.....	74
4.5.4.2.6	Calibration and time adjustment data .....	74
4.5.4.2.7	Events and faults data .....	75
4.5.4.2.8	Driver activity data .....	75
4.5.4.2.9	Vehicles used data .....	75
4.5.4.2.10	Daily work periods start and/or end data .....	75
4.5.4.2.11	Card session data .....	75
4.5.4.2.12	Control activity data .....	75
4.5.4.2.13	Vehicle units used data .....	75
4.5.4.2.14	Three hours continuous driving places data.....	76
4.5.4.2.15	Specific conditions data.....	76
4.5.5	Control card .....	76
4.5.5.1	Tachograph application (accessible to first and second generation vehicle units) .....	76
4.5.5.1.1	Application identification .....	76
4.5.5.1.2	Keys and Certificates.....	76
4.5.5.1.3	Card identification .....	76
4.5.5.1.4	Card holder identification .....	76
4.5.5.1.5	Control activity data .....	77
4.5.5.2	Tachograph G2 application (not accessible to first generation vehicle unit) .....	77
4.5.5.2.1	Application identification .....	77

4.5.5.2.2	Keys and Certificates.....	77
4.5.5.2.3	Card identification .....	77
4.5.5.2.4	Card holder identification .....	77
4.5.5.2.5	Control activity data .....	78
4.5.6 ..	Company card .....	78
4.5.6.1	Tachograph application (accessible to first and second generation vehicle units) .....	78
4.5.6.1.1	Application identification .....	78
4.5.6.1.2	Keys and Certificates.....	78
4.5.6.1.3	Card identification .....	78
4.5.6.1.4	Card holder identification .....	78
4.5.6.1.5	Company activity data .....	78
4.5.6.2	Tachograph G2 application (not accessible to first generation vehicle unit) .....	79
4.5.6.2.1	Application identification .....	79
4.5.6.2.2	Keys and Certificates.....	79
4.5.6.2.3	Card identification .....	79
4.5.6.2.4	Card holder identification .....	79
4.5.6.2.5	Company activity data .....	79
<b>5</b>	<b>INSTALLATION OF RECORDING EQUIPMENT .....</b>	<b>79</b>
<b>5.1</b>	<b>Installation .....</b>	<b>79</b>
<b>5.2</b>	<b>Installation plaque .....</b>	<b>81</b>
<b>5.3</b>	<b>Sealing .....</b>	<b>82</b>
<b>6</b>	<b>CHECKS, INSPECTIONS AND REPAIRS .....</b>	<b>83</b>
<b>6.1</b>	<b>Approval of fitters, workshops and vehicle manufacturers .....</b>	<b>83</b>
<b>6.2</b>	<b>Check of new or repaired instruments .....</b>	<b>83</b>
<b>6.3</b>	<b>Installation inspection .....</b>	<b>84</b>
<b>6.4</b>	<b>Periodic inspections .....</b>	<b>84</b>
<b>6.5</b>	<b>Measurement of errors .....</b>	<b>85</b>
<b>6.6</b>	<b>Repairs .....</b>	<b>85</b>
<b>7</b>	<b>CARD ISSUING .....</b>	<b>85</b>
<b>8</b>	<b>TYPE APPROVAL OF RECORDING EQUIPMENT AND TACHOGRAPH CARDS .....</b>	<b>86</b>
<b>8.1</b>	<b>General points .....</b>	<b>86</b>
<b>8.2</b>	<b>Security certificate .....</b>	<b>87</b>

<b>8.3</b>	<b>Functional certificate</b>	<b>87</b>
<b>8.4</b>	<b>Interoperability certificate</b>	<b>88</b>
<b>8.5</b>	<b>Type approval certificate</b>	<b>89</b>
<b>8.6</b>	<b>Exceptional procedure: first interoperability certificates for 2nd generation recording equipment and tachograph cards</b>	<b>89</b>



## INTRODUCTION

First generation digital tachograph system is deployed since 1 May 2006. It may be used until its end of life for domestic transportation. For international transportation instead, 15 years after the entry into force of this Commission Regulation, all vehicles shall be equipped with a compliant second generation smart tachograph, introduced by this Regulation.

This Annex contains second generation recording equipment and tachograph cards requirements.

Starting from its introduction date, second generation recording equipment shall be installed in vehicles registered for the first time, and second generation tachograph cards shall be issued.

In order to foster a smooth introduction of the second generation tachograph system,

- second generation tachograph cards shall be designed to be also used in first generation vehicle units,
- replacement of valid first generation tachograph cards at the introduction date shall not be requested.

This will allow drivers to keep their unique driver card and use both systems with it.

Second generation recording equipment shall however only be calibrated using second generation workshop cards.

This Annex contains all requirements related to the interoperability between the first and the second generation tachograph system.

Appendix 15 contains additional details about how the co-existence of the two systems shall be managed.

List of Appendixes

- App 1: DATA DICTIONARY
- App 2: TACHOGRAPH CARDS SPECIFICATION
- App 3: PICTOGRAMS
- App 4: PRINTOUTS
- App 5: DISPLAY
- App 6: FRONT CONNECTOR FOR CALIBRATION AND DOWNLOAD
- App 7: DATA DOWNLOADING PROTOCOLS
- App 8: CALIBRATION PROTOCOL
- App 9: TYPE APPROVAL AND LIST OF MINIMUM REQUIRED TESTS
- App 10: SECURITY REQUIREMENTS
- App 11: COMMON SECURITY MECHANISMS

- App 12: POSITIONING BASED ON GLOBAL NAVIGATION SATELLITE SYSTEM (GNSS)
- App 13: ITS INTERFACE
- App 14: REMOTE COMMUNICATION FUNCTION
- App 15: MIGRATION: MANAGING THE CO-EXISTENCE OF EQUIPMENT GENERATIONS
- App 16: ADAPTOR FOR M 1 AND N1 CATEGORY VEHICLES

## Definitions

In this Annex:

**a) “activation” means:**

the phase in which the tachograph becomes fully operational and implements all functions, including security functions, through the use of a workshop card;

**b) “authentication” means:**

a function intended to establish and verify a claimed identity;

**c) “authenticity” means:**

the property that information is coming from a party whose identity can be verified;

**d) “built-in-test (BIT)” means:**

tests run at request, triggered by the operator or by external equipment;

**e) “calendar day” means:**

a day ranging from 00.00 hours to 24.00 hours. All calendar days relate to UTC time (Universal Time Co-ordinated);

**f) “calibration” of a smart tachograph means:**

updating or confirming vehicle parameters to be held in the data memory. Vehicle parameters include vehicle identification (VIN, VRN and registering Member State) and vehicle characteristics (w, k, l, tyre size, speed limiting device setting (if applicable), current UTC time, current odometer value); during the calibration of a recording equipment, the types and identifiers of all type approval relevant seals in place shall also be stored in the data memory;

any update or confirmation of UTC time only, shall be considered as a time adjustment and not as a calibration, provided it does not contradict Requirement 409;

*calibrating a recording equipment requires the use of a workshop card;*

**g) “card number” means:**

a 16 alpha-numerical characters number that uniquely identifies a tachograph card within a Member State. The card number includes a card consecutive index (if applicable), a card replacement index and a card renewal index;

a card is therefore uniquely identified by the code of the issuing Member State and the card number;

**h) “card consecutive index” means:**

the 14<sup>th</sup> alpha-numerical character of a card number that is used to differentiate the different cards issued to a company, a workshop or a control authority entitled to be issued several tachograph cards. The company, the workshop or the control authority is uniquely identified by the 13 first characters of the card number;

**i) “card renewal index” means:**

the 16<sup>th</sup> alpha-numerical character of a card number which is incremented each time a tachograph card is renewed;

**j) “card replacement index” means:**

the 15<sup>th</sup> alpha-numerical character of a card number which is incremented each time a tachograph card is replaced;

**k) “characteristic coefficient of the vehicle” means:**

the numerical characteristic giving the value of the output signal emitted by the part of the vehicle linking it with the recording equipment (gearbox output shaft or axle) while the vehicle travels a distance of one kilometre under standard test conditions as defined under requirement 414. The characteristic coefficient is expressed in impulses per kilometre ( $w = \dots \text{imp/km}$ );

**l) “company card” means:**

a tachograph card issued by the authorities of a Member State to a transport undertaking needing to operate vehicles fitted with a tachograph, which identifies the transport undertaking and allows for the displaying, downloading and printing of the data, stored in the tachograph, which have been locked by that transport undertaking;

**m) “constant of the recording equipment” means:**

the numerical characteristic giving the value of the input signal required to show and record a distance travelled of one kilometre; this constant shall be expressed in impulses per kilometre ( $k = \dots \text{imp/km}$ );

**n) “continuous driving time” is computed within the recording equipment as<sup>1</sup>:**

the continuous driving time is computed as the current accumulated driving times of a particular driver, since the end of his last AVAILABILITY or BREAK/REST or UNKNOWN<sup>2</sup> period of 45 minutes or more (this period may have been split according to Regulation (EC) N°. 561/2006). The computations involved take into account, as needed, past activities stored on the driver card. When the driver has not inserted his card, the computations involved are based on the data memory recordings related to the current period where no card was inserted and related to the relevant slot;

**o) “control card” means:**

---

<sup>1</sup> This way of computing the continuous driving time and the cumulative break time serves into the Recording Equipment for computing the continuous driving time warning. It does not prejudice the legal interpretation to be made of these times. Alternative ways of computing the continuous driving time and the cumulative break time may be used to replace these definitions if they have been made obsolete by updates in other relevant legislation.

<sup>2</sup> UNKNOWN periods correspond to periods where the driver’s card was not inserted in a recording equipment and for which no manual entry of driver activities was made.

a tachograph card issued by the authorities of a Member State to a national competent control authority which identifies the control body and, optionally, the control officer, and which allows access to the data stored in the data memory or in the driver cards and, optionally, in the workshop cards for reading, printing and/or downloading;

It shall also give access to the roadside calibration checking function and to data on the remote early detection communication reader.

**p) “cumulative break time” is computed within the recording equipment as<sup>1</sup>:**

the cumulative break from driving time is computed as the current accumulated AVAILABILITY or BREAK/REST or UNKNOWN<sup>2</sup> times of 15 minutes or more of a particular driver, since the end of his last AVAILABILITY or BREAK/REST or UNKNOWN<sup>2</sup> period of 45 minutes or more (this period may have been split according to Regulation (EC) N°. 561/2006).

The computations involved take into account, as needed, past activities stored on the driver card. Unknown periods of negative duration (start of unknown period > end of unknown period) due to time overlaps between two different recording equipments, are not taken into account for the computation.

When the driver has not inserted his card, the computations involved are based on the data memory recordings related to the current period where no card was inserted and related to the relevant slot

**q) “data memory” means:**

an electronic data storage device built into the recording equipment;

**r) “digital signature” means:**

data appended to, or a cryptographic transformation of, a block of data that allows the recipient of the block of data to prove the authenticity and integrity of the block of data;

**s) “downloading” means:**

the copying, together with the digital signature, of a part, or of a complete set, of data files recorded in the data memory of the vehicle unit or in the memory of a tachograph card, provided that this process does not alter or delete any stored data;

Manufacturers of smart tachograph vehicle units and manufacturers of equipment designed and intended to download data files shall take all reasonable steps to ensure that the downloading of such data can be performed with the minimum delay by transport undertakings or drivers.

The downloading of the detailed speed file may not be necessary to establish compliance with Regulation (EC) N°. 561/2006, but may be used for other purposes such as accident investigation.

**t) “driver card” means:**

a tachograph card, issued by the authorities of a Member State to a particular driver, which identifies the driver and allows for the storage of driver activity data;

**u) “effective circumference of the wheels” means:**

the average of the distances travelled by each of the wheels moving the vehicle (driving wheels) in the course of one complete rotation. The measurement of these distances shall be made under standard test conditions as defined under requirement 414 and is expressed in the form “l = ... mm”. Vehicle manufacturers may replace the measurement of these distances by a theoretical calculation which takes into account the distribution of the weight on the axles, vehicle unladen in normal running order<sup>3</sup>. The methods for such theoretical calculation are subject to approval by a competent Member State authority and can take place only before tachograph activation;

**v) “event” means:**

an abnormal operation detected by the smart tachograph which may result from a fraud attempt;

**w) “external GNSS facility” means**

a facility which contains the GNSS receiver when the vehicle unit is not a single unit as well as other components needed to protect the communication of position data to the rest of the vehicle unit;

**x) “fault” means:**

abnormal operation detected by the smart tachograph which may come from an equipment malfunction or failure;

**y) “GNSS receiver” means:**

an electronic device that receives and digitally processes the signals from one or more Global Navigation Satellite System(s) (GNSS in English) in order to provide position, speed and time information.

**z) “installation” means:**

the mounting of a tachograph in a vehicle;

**aa) ‘interoperability’ means:**

the capacity of systems and the underlying business processes to exchange data and to share information;

**bb) ‘interface’ means:**

a facility between systems which provides the media through which they can connect and interact;

**cc) “position” means:**

geographical coordinates of the vehicle at a given time;

---

<sup>3</sup> Regulation (EU) N°. 1230/2012 relating to the masses and dimensions of certain categories of motor vehicles and their trailers and amending Directive 2007/46/EC, as last amended.

**dd) “motion sensor” means:**

a part of the tachograph, providing a signal representative of vehicle speed and/or distance travelled;

**ee) “non valid card” means:**

a card detected as faulty, or which initial authentication failed, or whose start of validity date is not yet reached, or which expiry date has passed;

**ff) ‘open standard’ means:**

a standard set out in a standard specification document available freely or at a nominal charge which it is permissible to copy, distribute or use for no fee or for a nominal fee.

**gg) “out of scope” means:**

when the use of the recording equipment is not required, according to the provisions of Regulation (EC) N°. 561/2006.

**hh) “over speeding” means:**

exceeding the authorised speed of the vehicle, defined as any period of more than 60 seconds during which the vehicle’s measured speed exceeds the limit for setting the speed limitation device laid down in Council Directive 92/6/EEC of 10 February 1992 on the installation and use of speed limitation devices for certain categories of motor vehicles in the Community<sup>4</sup>, as last amended;

**ii) “periodic inspection” means:**

a set of operations performed to check that the tachograph works properly, that its settings correspond to the vehicle parameters, and that no manipulation devices are attached to the tachograph;

**jj) “printer” means:**

component of the recording equipment which provides printouts of stored data;

**kk) “remote early detection communication” means:**

communication between the remote early detection communication facility and the remote early detection communication reader during targeted roadside checks with the aim of remotely detecting possible manipulation or misuse of recording equipment ;

**ll) “remote communication facility” means:**

---

<sup>4</sup> OJ No L 057, 02/03/1992, p. 0027 - 0028

the equipment of the vehicle unit which is used to perform targeted roadside checks;

**mm) “remote early detection communication reader” means:**

the system used by control officers for targeted roadside checks.

**nn) “renewal” means:**

issue of a new tachograph card when an existing card reaches its expiry date, or is malfunctioning and has been returned to the issuing authority. Renewal always implies the certainty that two valid cards do not co-exist;

**oo) “repair” means:**

any repair of a motion sensor or of a vehicle unit or of a cable that requires the disconnection of its power supply, or its disconnection from other tachograph components, or the opening of the motion sensor or vehicle unit;

**pp) “card replacement” means:**

issue of a tachograph card in replacement of an existing card, which has been declared lost, stolen or malfunctioning and has not been returned to the issuing authority. Replacement always implies a risk that two valid cards may co-exist;

**qq) “security certification” means:**

process to certify, by a Common Criteria certification body, that the recording equipment (or component) or the tachograph card under investigation fulfils the security requirements defined in the relative Protection Profiles;

**rr) “self test” means:**

tests run cyclically and automatically by the recording equipment to detect faults;

**ss) “time measurement” means:**

a permanent digital record of the coordinated universal date and time (UTC);

**tt) “time adjustment” means:**

an automatic adjustment of current time at regular intervals and within a maximum tolerance of one minute, or an adjustment performed during calibration;

**uu) “tyre size” means:**



the designation of the dimensions of the tyres (external driving wheels) in accordance with Directive 92/23/EEC of 31 March 1992<sup>5</sup> as last amended;

**vv) “vehicle identification” means:**

numbers identifying the vehicle: Vehicle Registration Number (VRN) with indication of the registering Member State and Vehicle Identification Number (VIN)<sup>6</sup>;

**ww) for computing sake in the recording equipment “week” means:**

the period between 00.00 hours UTC on Monday and 24.00 UTC on Sunday;

**xx) “workshop card” means:**

a tachograph card issued by the authorities of a Member State to designated staff of a tachograph manufacturer, a fitter, a vehicle manufacturer or a workshop, approved by that Member State, which identifies the cardholder and allows for the testing, calibration and activation of tachographs, and/or downloading from them;

**yy) “adaptor” means:**

a device, providing a signal permanently representative of vehicle speed and/or distance travelled, other than the one used for the independent movement detection, and which is:

- installed and used only in M1 and N1 type vehicles (as defined in Annex II to Council Directive 2007/46/EC , as last amended) put into service since 1 May 2006,
- installed where it is not mechanically possible to install any other type of existing motion sensor which is otherwise compliant with the provisions of this Annex and its Appendixes 1 to 15,
- installed between the vehicle unit and where the speed/distance impulses are generated by integrated sensors or alternative interfaces,
- seen from a vehicle unit, the adaptor behaviour is the same as if a motion sensor, compliant with the provisions of this Annex and its Appendixes 1 to 16, was connected to the vehicle unit;

use of such an adaptor in those vehicles described above shall allow for the installation and correct use of a vehicle unit compliant with all the requirements of this Annex,

for those vehicles, the smart tachograph includes cables, an adaptor, and a vehicle unit;

**zz) data integrity means:**

the accuracy and consistency of stored data, indicated by an absence of any alteration in data between two updates of a data record. Integrity implies that the data is an exact copy of the original version, e.g. that it has not been corrupted in the process of being written to, and read back from, a tachograph card or a dedicated equipment or during transmission via any communications channel;

**aaa) data privacy means:**

---

<sup>5</sup> OJ No L 129, 14/05/1992, p. 0095.

<sup>6</sup> Directive 76/114/EEC, 18/12/1975; OJ No L 024, 30/01/1976, p. 0001 - 0005.

the overall technical measures taken to ensure the proper implementation of the principles laid down in Directive 95/46/EC of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data as well as of those laid down in Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector;

**bbb) smart tachograph system means:**

the recording equipment, tachograph cards and the set of all directly or indirectly interacting equipment during their construction, installation, use, testing and control, such as cards, remote communication reader and any other equipment for data downloading, data analysis, calibration, generating, managing or introducing security elements, etc;

**ccc) introduction date:**

36 months after the entry into force of the detailed provisions referred to in Article 11 of Regulation (EU) N°. 165/2014.

*This is the date after which vehicles registered for the first time:*

- *shall be fitted with a tachograph connected to a positioning service based on a satellite navigation system,*
- *shall be able to communicate data for targeted roadside checks to competent control authorities while the vehicle is in motion,*
- *and may be equipped with standardised interfaces allowing the data recorded or produced by tachographs to be used in operational mode, by an external device.*

**ddd) protection profile means:**

a document used as part of certification process according Common Criteria, providing implementation independent specification of information assurance security requirements;

**eee) GNSS accuracy:**

in the context of recording the position from Global Navigation Satellite System (GNSS) with tachographs, means the value of the Horizontal Dilution of Precision (HDOP) calculated as the minimum of the HDOP values collected on the available GNSS systems.

1 General characteristics and functions of the recording equipment

1.1 General characteristics

The purpose of the recording equipment is to record, store, display, print, and output data related to driver activities.

Any vehicle fitted with the recording equipment complying with the provisions of this Annex, must include a speed display and an odometer. These functions may be included within the recording equipment.

- 01) The recording equipment includes cables, a motion sensor, and a vehicle unit.
- 02) The interface between motion sensors and vehicle units shall comply with the requirements specified in Appendix 11.
- 03) The vehicle unit shall be connected to global navigation satellite system(s), as specified in Appendix 12.
- 04) The vehicle unit shall communicate with remote early detection communication readers, as specified in Appendix 14.
- 05) The vehicle unit may include an ITS interface, which is specified in Appendix 13  
The recording equipment may be connected to other facilities through additional interfaces and/or through the optional ITS interface.
- 06) Any inclusion in or connection to the recording equipment of any function, device, or devices, approved or otherwise, shall not interfere with, or be capable of interfering with, the proper and secure operation of the recording equipment and the provisions of this Regulation.

Recording equipment users identify themselves to the equipment via tachograph cards.

- 07) The recording equipment provides selective access rights to data and functions according to user's type and/or identity.

The recording equipment records and stores data in its data memory, in the remote communication facility and in tachograph cards.

This is done in accordance with Directive 95/46/EC of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data<sup>7</sup>, with Directive 2002/58/EC of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector<sup>8</sup> and in compliance with Article 7 of Regulation (EU) N°. 165/2014.

## 1.2 Functions

- 08) The recording equipment shall ensure the following functions:
- monitoring cards insertions and withdrawals,
  - speed, distance and position measurement,

---

<sup>7</sup> OJ No L 281, 23/11/1995, p.31.

<sup>8</sup> OJ N° L 201, 31/07/2002, p.37

- time measurement,
- monitoring driver activities,
- monitoring driving status,
- drivers manual entries:
  - o entry of places where daily work periods begin and/or end,
  - o manual entry of driver activities,
  - o entry of specific conditions,
- company locks management,
- monitoring control activities,
- detection of events and/or faults,
- built-in and self-tests,
- reading from data memory,
- recording and storing in data memory,
- reading from tachograph cards,
- recording and storing in tachograph cards,
- displaying,
- printing,
- warning,
- data downloading to external media,
- remote communication for targeted roadside checks,
- output data to additional facilities,
- calibration,
- roadside calibration check,
- time adjustment.

### 1.3 Modes of operation

- 09) The recording equipment shall possess four modes of operation:
- operational mode,
  - control mode,
  - calibration mode,
  - company mode.
- 10) The recording equipment shall switch to the following mode of operation according to the valid tachograph cards inserted into the card interface devices. In order to determine the mode of operation, the tachograph card generation is

irrelevant, provided the inserted card is valid. A first generation workshop card shall always be considered as non-valid when it is inserted in a second generation VU.

Mode of operation		Driver slot				
		No card	Driver card	Control card	Workshop card	Company card
Co-driver slot	No card	Operational	Operational	Control	Calibration	Company
	Driver card	Operational	Operational	Control	Calibration	Company
	Control card	Control	Control	Control (*)	Operational	Operational
	Workshop card	Calibration	Calibration	Operational	Calibration (*)	Operational
	Company card	Company	Company	Operational	Operational	Company (*)

(\*) In these situations the recording equipment shall use only the tachograph card inserted in the driver slot.

- 11) The recording equipment shall ignore non valid cards inserted, except displaying, printing or downloading data held on an expired card which shall be possible.
- 12) All functions listed in 1.2. shall work in any mode of operation with the following exceptions:
  - the calibration function is accessible in the calibration mode only,
  - the roadside calibration checking function is accessible in the control mode only,
  - the company locks management function is accessible in the company mode only,
  - the monitoring of control activities function is operational in the control mode only,
  - The downloading function is not accessible in the operational mode (except as provided for in requirement 193), and except downloading a driver card when no other card type is inserted into the VU.
- 13) The recording equipment can output any data to display, printer or external interfaces with the following exceptions:
  - in the operational mode, any personal identification (surname and first name(s)) not corresponding to a tachograph card inserted shall be blanked and any card number not corresponding to a tachograph card inserted shall be partially blanked (every odd character – from left to right - shall be blanked),
  - in the company mode, driver related data (requirements 102, 105 and 108) can be output only for periods where no lock exists or no other company holds a lock (as identified by the first 13 digits of the company card number),
  - when no card is inserted in the recording equipment, driver related data can be output only for the current and 8 previous calendar days,
  - personal data originating from the VU shall not be output through ITS interface of the VU unless the consent of the driver to whom the data relates is verified,
  - the vehicle units have a normal operations validity period of 15 years, starting with the vehicle unit certificates issuing date, but vehicle units can be used for additional 3 months, for data downloading only.

#### 1.4 Security

The system security aims at protecting the data memory in such a way as to prevent unauthorised access to and manipulation of the data and detecting any such attempts, protecting the integrity and authenticity of data exchanged between the motion sensor and the vehicle unit, protecting the integrity and authenticity of data exchanged between the recording equipment and the tachograph cards, protecting the integrity and authenticity of data exchanged between the recording equipment and the external GNSS facility, protecting the confidentiality, integrity and authenticity of data exchanged through the remote early detection communication for control purposes, and verifying the integrity and authenticity of data downloaded.

- 14) In order to achieve the system security, the following components shall meet the security requirements specified in their Protection Profiles, as required in Appendix 10:
- vehicle unit,
  - tachograph card,
  - motion sensor,
  - external GNSS facility (this Profile is only needed and applicable for the external GNSS variant).

#### 2 Construction and functional requirements for recording equipment

##### 2.1 Monitoring cards insertion and withdrawal

- 15) The recording equipment shall monitor the card interface devices to detect card insertions and withdrawals.
- 16) Upon card insertion the recording equipment shall detect whether the card inserted is a valid tachograph card and in such a case identify the card type and the card generation.
- If a card with the same card number and a higher renewal index has already been inserted in the recording equipment, the card shall be declared as non-valid.
- If a card with the same card number and renewal index but with a higher replacement index has already been inserted in the recording equipment, the card shall be declared as non-valid.
- 17) First generation tachograph cards shall be considered as non-valid by the recording equipment, after the possibility of using first generation tachograph cards has been suppressed by a workshop, in compliance with Appendix 15 (req. MIG003).
- 18) First generation workshop cards which are inserted in the second generation recording equipment shall be considered as non-valid.
- 19) The recording equipment shall be so designed that the tachograph cards are locked in position on their proper insertion into the card interface devices.
- 20) The release of tachograph cards may function only when the vehicle is stopped and after the relevant data have been stored on the cards. The release of the card shall require positive action by the user.

## 2.2 Speed, position and distance measurement

- 21) The motion sensor (possibly embedded in the adaptor) is the main source for speed and distance measurement.
- 22) This function shall continuously measure and be able to provide the odometer value corresponding to the total distance travelled by the vehicle using the pulses provided by the motion sensor.
- 23) This function shall continuously measure and be able to provide the speed of the vehicle using the pulses provided by the motion sensor.
- 24) The speed measurement function shall also provide the information whether the vehicle is moving or stopped. The vehicle shall be considered as moving as soon as the function detects more than 1 imp/sec for at least 5 seconds from the motion sensor, otherwise the vehicle shall be considered as stopped.
- 25) Devices displaying speed (speedometer) and total distance travelled (odometer) installed in any vehicle fitted with a recording equipment complying with the provisions of this Regulation, shall comply with the requirements relating to maximum tolerances (see 3.2.1 and 3.2.2) laid down in this Annex.
- 26) To detect manipulation of motion data, information from the motion sensor shall be corroborated by vehicle motion information derived from the GNSS receiver and optionally by other source(s) independent from the motion sensor.
- 27) This function shall measure the position of the vehicle in order to allow for the automatic recording of:
  - positions where the driver and/or the co-driver begins his daily work period;
  - positions where the continuous driving time of the driver reaches a multiple of three hours;
  - positions where the driver and/or the co-driver ends his daily work period.

### 2.2.1 Measurement of distance travelled

- 28) The distance travelled may be measured either:
  - so as to cumulate both forward and reverse movements, or
  - so as to include only forward movement.
- 29) The recording equipment shall measure distance from 0 to 9 999 999.9 km.
- 30) Distance measured shall be within the following tolerances (distances of at least 1000 m.):
  - $\pm 1\%$  before installation,

- $\pm 2\%$  on installation and periodic inspection,
- $\pm 4\%$  in use.

31) Distance measured shall have a resolution better than or equal to 0.1 km.

#### 2.2.2 Measurement of speed

32) The recording equipment shall measure speed from 0 to 220 km/h.

33) To ensure a maximum tolerance on speed displayed of  $\pm 6$  km/h in use, and taking into account:

- a  $\pm 2$  km/h tolerance for input variations (tyre variations, ...),
- a  $\pm 1$  km/h tolerance in measurements made during installation or periodic inspections,

the recording equipment shall, for speeds between 20 and 180 km/h, and for characteristic coefficients of the vehicle between 4000 and 25000 imp/km, measure the speed with a tolerance of  $\pm 1$  km/h (at constant speed).

Note: The resolution of data storage brings an additional tolerance of  $\pm 0.5$  km/h to speed stored by the recording equipment.

34) The speed shall be measured correctly within the normal tolerances within 2 seconds of the end of a speed change when the speed has changed at a rate up to  $2\text{m/s}^2$ .

35) Speed measurement shall have a resolution better than or equal to 1 km/h.

#### 2.2.3 Measurement of position

36) The recording equipment shall measure the absolute position of the vehicle using the GNSS receiver.

37) The absolute position is measured in geographical coordinates of latitude and longitude in degrees and minutes with a resolution of 1/10 of a minute.

#### 2.3 Time measurement

38) The time measurement function shall measure permanently and digitally provide UTC date and time.

39) UTC date and time shall be used for dating data inside the recording equipment (recordings, data exchange) and for all printouts specified in Appendix 4 "Printouts".

40) In order to visualise the local time, it shall be possible to change the offset of the time displayed, in half hour steps. No other offsets than negative or positive multiples of half hours shall be allowed;

41) Time drift shall be within  $\pm 2$  seconds per day in type approval conditions, in the absence of any time adjustment.



- 42) Time measured shall have a resolution better than or equal to 1 second.
- 43) Time measurement shall not be affected by an external power supply cut-off of less than 12 months in type approval conditions.

#### 2.4 Monitoring driver activities

- 44) This function shall permanently and separately monitor the activities of one driver and one co-driver.
- 45) Driver activity shall be DRIVING, WORK, AVAILABILITY or BREAK/REST.
- 46) It shall be possible for the driver and/or the co-driver to manually select WORK, AVAILABILITY or BREAK/REST.
- 47) When the vehicle is moving, DRIVING shall be selected automatically for the driver and AVAILABILITY shall be selected automatically for the co-driver.
- 48) When the vehicle stops, WORK shall be selected automatically for the driver.
- 49) The first change of activity to REST or AVAILABILITY arising within 120 seconds of the automatic change to WORK due to the vehicle stop shall be assumed to have happened at the time of vehicle stop (therefore possibly cancelling the change to WORK).
- 50) This function shall output activity changes to the recording functions at a resolution of one minute.
- 51) Given a calendar minute, if DRIVING is registered as the activity of both the immediately preceding and the immediately succeeding minute, the whole minute shall be regarded as DRIVING.
- 52) Given a calendar minute that is not regarded as DRIVING according to requirement 051, the whole minute shall be regarded to be of the same type of activity as the longest continuous activity within the minute (or the latest of the equally long activities).
- 53) This function shall also permanently monitor the continuous driving time and the cumulative break time of the driver.

#### 2.5 Monitoring driving status

- 54) This function shall permanently and automatically monitor the driving status.

- 55) The driving status CREW shall be selected when two valid driver cards are inserted in the equipment, the driving status SINGLE shall be selected in any other case.

## 2.6 Drivers entries

### 2.6.1 Entry of places where daily work periods begin and/or end

- 57) This function shall allow for the entry of places where, according to the driver and/or the co-driver, his daily work periods begin and/or end.
- 57) Places are defined as the country and, in addition where applicable, the region, which are entered or confirmed manually.
- 58) At the time of a driver card withdrawal, the recording equipment shall prompt the (co-)driver to enter a “place where the daily work period ends”.
- 59) The driver shall then enter the current place of the vehicle, which shall be considered as a temporary entry.
- 60) It shall be possible to input places where daily work periods begin and/or end through commands in the menus. If more than one such input is done within one calendar minute, only the last begin place input and the last end place input done within that time shall be kept recorded.

### 2.6.2 Manual entry of driver activities and driver consent for ITS interface

- 61) Upon driver (or workshop) card insertion, and only at this time, the recording equipment shall allow manual entries of activities. Manual entries of activities shall be performed using local time and date values of the time zone (UTC offset) currently set for the vehicle unit.

At driver or workshop card insertion the cardholder shall be reminded of:

- the date and time of his last card withdrawal;
- optionally: the local time offset currently set for the vehicle unit.

At the first insertion of a given driver card or workshop card currently unknown to the vehicle unit, the cardholder shall be invited to express his consent for tachograph related personal data output through the optional ITS interface.

At any moment, the driver (resp. workshop) consent can be enabled or disabled through commands in the menu, provided the driver (resp. workshop) card is inserted.

It shall be possible to input activities with the following restrictions:

- Activity type shall be WORK, AVAILABILITY or BREAK/REST;
- Start and end times for each activity shall be within the period of the last card withdrawal – current insertion only;
- Activities shall not be allowed to overlap mutually in time.

It shall be possible to make manual entries, if required, at the first insertion of a previously unused driver (or workshop) card.

The procedure for manual entries of activities shall include as many consecutive steps as necessary to set a type, a start time and an end time for each activity. For any part of the time period between last card withdrawal and current card insertion, the cardholder shall have the option not to declare any activity.

During the manual entries associated with card insertion and if applicable, the card holder shall have the opportunity to input:

- a place where a previous daily work period ended, associated to the relevant time (thus overwriting the entry made at the last card withdrawal),
- a place where the current daily work period begins, associated to the relevant time.

If the card holder doesn't enter any place where the work period begins or ended, during the manual entries associated with card insertion, this shall be considered as a declaration that his work period has not changed since the last card withdrawal. The next entry of a place where a previous daily work period ends shall then overwrite the temporary entry made at the last card withdrawal.

If a place is entered, it shall be recorded in the relevant tachograph card.

Manual entries shall be interrupted if:

- the card is withdrawn or,
- the vehicle is moving and the card is in the driver slot.

Additional interruptions are allowed, e.g. a timeout after a certain period of user inactivity. If manual entries are interrupted, the recording equipment shall validate any complete place and activity entries (having either unambiguous place and time, or activity type, begin time and end time) already made.

If a second driver or workshop card is inserted while manual entries of activities are in progress for a previously inserted card, the manual entries for this previous card shall be allowed to be completed before manual entries start for the second card.

The cardholder shall have the option to insert manual entries according to the following minimum procedure:

- Enter activities manually, in chronological order, for the period last card withdrawal – current insertion.
- Begin time of the first activity shall be set to card withdrawal time. For each subsequent entry, the start time shall be preset to immediately follow the end time of the previous entry. Activity type and end time shall be selected for each activity.

The procedure shall end when the end time of a manually entered activity equals the card insertion time. The recording equipment may then optionally allow the card holder to modify any activity manually entered, until validation by selection of a specific command. Thereafter, any such modification shall be forbidden.

### 2.6.3 Entry of specific conditions

62) The recording equipment shall allow the driver to enter, in real time, the following two specific conditions:

- “OUT OF SCOPE” (begin, end)
- “FERRY / TRAIN CROSSING” (begin, end).

A “FERRY / TRAIN CROSSING” may not occur if an “OUT OF SCOPE” condition is opened.

An opened “OUT OF SCOPE” condition must be automatically closed, by the recording equipment, if a driver card is inserted or withdrawn.

An opened "OUT OF SCOPE" condition shall inhibit the following events and warnings:

- Driving without an appropriate card,
- Warnings associated with continuous driving time.

The FERRY / TRAIN CROSSING begin flag shall be set before shutting down the engine on the ferry/train.

An opened FERRY / TRAIN CROSSING must end when any of following options occurs:

- The driver manually ends the FERRY/TRAIN CROSSING
- The driver ejects his card

An opened FERRY/TRAIN CROSSING shall end when it is no longer valid based on the rules stated in Regulation (EC) N°. 561/2006.

### 2.7 Company locks management

63) This function shall allow the management of the locks placed by a company to restrict data access in company mode to itself.

- 64) Company locks consist in a start date/time (lock-in) and an end date/time (lock-out) associated with the identification of the company as denoted by the company card number (at lock-in).
- 65) Locks may be turned “in” or “out” in real time only.
- 66) Locking-out shall only be possible for the company whose lock is “in” (as identified by the first 13 digits of the company card number), or,
- 67) Locking-out shall be automatic if another company locks in.
- 68) In the case where a company locks in and where the previous lock was for the same company, then it will be assumed that the previous lock has not been turned “out” and is still “in”.

2.8 Monitoring control activities

- 69) This function shall monitor DISPLAYING, PRINTING, VU and card DOWNLOADING, and ROADSIDE CALIBRATION check activities carried while in control mode.
- 70) This function shall also monitor OVER SPEEDING CONTROL activities while in control mode. An over speeding control is deemed to have happened when, in control mode, the “over speeding” printout has been sent to the printer or to the display, or when “events and faults” data have been downloaded from the VU data memory.

2.9 Detection of events and/or faults

- 71) This function shall detect the following events and/or faults:

2.9.1 “Insertion of a non-valid card” event

- 72) This event shall be triggered at the insertion of any non-valid card, at the insertion of a driver card already replaced and/or when an inserted valid card expires.

2.9.2 “Card conflict” event

- 73) This event shall be triggered when any of the valid cards combination noted X in the following table arises:

Card conflict		Driver slot				
		No card	Driver card	Control card	Workshop card	Company card
Card conflict	No card					
	Driver card				X	
	Control card			X	X	X

	Workshop card		X	X	X	X
	Company card			X	X	X

2.9.3 “Time overlap” event

- 74) This event shall be triggered when the date / time of last withdrawal of a driver card, as read from the card, is later than the current date / time of the recording equipment in which the card is inserted.

2.9.4 “Driving without an appropriate card” event

- 75) This event shall be triggered for any valid tachograph cards combination noted X in the following table, when driver activity changes to DRIVING, or when there is a change of the mode of operation while driver activity is DRIVING:

Driving without an appropriate card		Driver slot				
		No (or non-valid) card	Driver card	Control card	Workshop card	Company card
Card in driver slot	No (or non-valid) card	X		X		X
	Driver card	X		X	X	X
	Control card	X	X	X	X	X
	Workshop card	X	X	X		X
	Company card	X	X	X	X	X

2.9.5 “Card insertion while driving” event

- 76) This event shall be triggered when a tachograph card is inserted in any slot, while driver activity is DRIVING.

2.9.6 “Last card session not correctly closed” event

- 77) This event shall be triggered when at card insertion the recording equipment detects that, despite the provisions laid down in paragraph 3.1., the previous card session has not been correctly closed (the card has been withdrawn before all relevant data have been stored on the card). This event shall be triggered by driver and workshop cards only.

2.9.7 “Over speeding” event

- 78) This event shall be triggered for each over speeding.

#### 2.9.8 “Power supply interruption” event

- 79) This event shall be triggered, while not in calibration or control mode, in case of any interruption exceeding 200 milliseconds of the power supply of the motion sensor and/or of the vehicle unit. The interruption threshold shall be defined by the manufacturer. The drop in power supply due to the starting of the engine of the vehicle shall not trigger this event.

#### ***3.9.9 “Communication error with the remote communication facility” event***

- 80) This event shall be triggered, **while not in calibration mode**, when the remote communication facility does not acknowledge the successful reception of remote communication data sent from the vehicle unit for more than three attempts.

#### ***3.9.10 “Absence of position information from GNSS receiver” event***

- 81) This event shall be triggered, **while not in calibration mode**, in case of absence of position information originating from the GNSS receiver (whether internal or external) for more than three hours of accumulated driving time.

#### ***3.9.11 “Communication error with the external GNSS facility” event***

- 82) This event shall be triggered, **while not in calibration mode**, in case of interruption of the communication between the external GNSS facility and the vehicle unit for more than 20 continuous minutes, when the vehicle is moving.

#### 3.9.12 “Motion data error” event

- 83) This event shall be triggered, **while not in calibration mode**, in case of interruption of the normal data flow between the motion sensor and the vehicle unit and/or in case of data integrity or data authentication error during data exchange between the motion sensor and the vehicle unit.

#### ***3.9.13 “Vehicle motion conflict” event***

- 84) This event shall be triggered, **while not in calibration mode**, in case motion information calculated from the motion sensor is contradicted by motion information calculated from the internal GNSS receiver or from the external GNSS facility and optionally by other independent sources, as specified in Appendix 12. This event shall not be triggered during a ferry/train crossing, an OUT OF SCOPE condition, or when the position information from the GNSS receiver is not available.

#### 3.9.14 “Security breach attempt” event

- 85) This event shall be triggered for any other event affecting the security of the motion sensor and/or of the vehicle unit and/or the external GNSS facility as required in Appendix 10, while not in calibration mode.

### 3.9.15: “Time conflict” event

- 86) This event shall be triggered, **while not in calibration mode**, when the VU detects a discrepancy of more than 1 minute between the time of the vehicle unit’s time measurement function and the time originating from the GNSS receiver. This event is recorded together with the internal clock value of the vehicle unit and comes together with an automatic time adjustment. After a time conflict event has been triggered, the VU will not generate other time conflict events for the next 12 hours. This event shall not be triggered in cases no valid GNSS signal was detectable by the GNSS receiver within the last 30 days. However, when the position information from the GNSS receiver is available again, the automatic time adjustment shall be done.

### 3.9.16 “Card” fault

- 87) This fault shall be triggered when a tachograph card failure occurs during operation.

### 3.9.17 “Recording equipment” fault

- 88) This fault shall be triggered for any of these failures, while not in calibration mode:
- VU internal fault
  - Printer fault
  - Display fault
  - Downloading fault
  - Sensor fault
  - GNSS receiver or external GNSS facility fault
  - Remote Communication facility fault

## 2.10 Built-in and self-tests

- 89) The recording equipment shall self-detect faults through self-tests and built-in-tests, according to the following table:

90)

<i>Sub-assembly to test</i>	<i>self-test</i>	<i>Built-in-test</i>
Software		Integrity
Data memory	Access	Access, data integrity
Card interface devices	Access	Access
Keyboard		Manual check
Printer	(up to manufacturer)	Printout



<i>Sub-assembly to test</i>	<i>self-test</i>	<i>Built-in-test</i>
Display		Visual check
Downloading (performed only during downloading)	Proper operation	
Sensor	Proper operation	Proper operation
Remote communication facility	Proper operation	Proper operation
GNSS facility	Proper operation	Proper operation

### 2.11 Reading from data memory

- 91) The recording equipment shall be able to read any data stored in its data memory.

### 2.12 Recording and storing in the data memory

For the purpose of this paragraph,

- “365 days” is defined as 365 calendar days of average drivers’ activity in a vehicle. The average activity per day in a vehicle is defined as at least 6 drivers or co-drivers, 6 card insertion withdrawal cycles, and 256 activity changes. “365 days” therefore include at least 2190 (co-)drivers, 2190 card insertion withdrawal cycles, and 93440 activity changes,
  - the average number of positions per day is defined as at least 6 positions where the daily work period begins, 6 positions when the driver’s continuous driving time reaches a multiple of three hours, and 6 positions where the daily work period ends, so that “365 days” include at least 6570 positions,
  - times are recorded with a resolution of one minute, unless otherwise specified,
  - odometer values are recorded with a resolution of one kilometre,
  - speeds are recorded with a resolution of 1 km/h,
  - positions (latitudes and longitudes) are recorded in degrees and minutes, with a resolution of 1/10 of minute, with the associated GNSS accuracy and acquisition time.
- 92) Data stored into the data memory shall not be affected by an external power supply cut-off of less than twelve months in type approval conditions. In addition, data stored in the external remote communication facility, as defined in Appendix 14, shall not be affected by power-supply cut-off of less than 28 days.
- 93) The recording equipment shall be able to record and store implicitly or explicitly in its data memory the following:
- 2.12.1 Equipment identification data
- 2.12.1.1 Vehicle Unit identification data
- 94) The recording equipment shall be able to store in its data memory the following vehicle unit identification data:

- name of the manufacturer,
- address of the manufacturer,
- part number,
- serial number,
- VU generation,
- ability to use first generation tachograph cards,
- software version number,
- software version installation date,
- year of equipment manufacture,
- approval number,

95) Vehicle unit identification data are recorded and stored once and for all by the vehicle unit manufacturer, except the software related data and the approval number which may be changed in case of software upgrade and the ability to use first generation tachograph cards.

#### 2.12.1.2 Motion sensor identification data

96) The motion sensor shall be able to store in its memory the following identification data:

- name of the manufacturer,
- serial number,
- approval number,
- embedded security component identifier (e.g. internal chip/processor part number),
- operating system identifier (e.g. software version number).

97) Motion sensor identification data are recorded and stored once and for all in the motion sensor, by the motion sensor manufacturer.

98) The vehicle unit shall be able to record and store in its data memory the following data related to the 20 most recent pairing of motion sensors (if several pairings happen within one calendar day, only the first and the last one of the day shall be stored):

The following data shall be recorded for each of these pairings:

- motion sensor identification data:
  - o serial number
  - o approval number
- motion sensor pairing data:

- pairing date.

#### 2.12.1.3 Global Navigation Satellite Systems identification data

99) The external GNSS facility shall be able to store in its memory the following identification data:

- name of the manufacturer,
- serial number,
- approval number,
- embedded security component identifier (e.g. internal chip/processor part number),
- operating system identifier (e.g. software version number).

100) The identification data are recorded and stored once and for all in the external GNSS facility, by the external GNSS facility manufacturer.

101) The vehicle unit shall be able to record and store in its data memory the following data related to the 20 most recent couplings of external GNSS facilities (if several couplings happen within one calendar day, only the first and the last one of the day shall be stored).

The following data shall be recorded for each of these couplings:

- external GNSS facility identification data:
  - serial number,
  - approval number,
- external GNSS facility coupling data:
  - coupling date

#### 2.12.2 Keys and Certificates

102) The recording equipment shall be able to store a number of cryptographic keys and certificates, as specified in Appendix 11 part A and part B.

#### 2.12.3 Driver or workshop card insertion and withdrawal data

103) For each insertion and withdrawal cycle of a driver or workshop card in the equipment, the recording equipment shall record and store in its data memory:

- the card holder's surname and first name(s) as stored in the card,
- the card's number, issuing Member State and expiry date as stored in the card,
- the card generation,
- the insertion date and time,
- the vehicle odometer value at card insertion,

- the slot in which the card is inserted,
- the withdrawal date and time,
- the vehicle odometer value at card withdrawal,
- the following information about the previous vehicle used by the driver, as stored in the card:
  - VRN and registering Member State,
  - VU generation (when available),
  - card withdrawal date and time,
- a flag indicating whether, at card insertion, the card holder has manually entered activities or not.

104) The data memory shall be able to hold these data for at least 365 days.

105) When storage capacity is exhausted, new data shall replace oldest data.

#### 2.12.4 Driver activity data

106) The recording equipment shall record and store in its data memory whenever there is a change of activity for the driver and/or the co-driver, and/or whenever there is a change of driving status, and/or whenever there is an insertion or withdrawal of a driver or workshop card:

- the driving status (CREW, SINGLE),
- the slot (DRIVER, CO-DRIVER),
- the card status in the relevant slot (INSERTED, NOT INSERTED),
- the activity (DRIVING, AVAILABILITY, WORK, BREAK/REST),
- the date and time of the change.

INSERTED means that a valid driver or workshop card is inserted in the slot. NOT INSERTED means the opposite i.e. no valid driver or workshop card is inserted in the slot (e.g. a company card is inserted or no card is inserted)

Activity data manually entered by a driver are not recorded in the data memory.

107) The data memory shall be able to hold driver activity data for at least 365 days.

108) When storage capacity is exhausted, new data shall replace oldest data.

#### 2.12.5 Places and positions where daily work periods begin, end, and/or where 3 hours continuous driving time is reached

109) The recording equipment shall record and store in its data memory:

- places and positions where the driver and/or the co-driver begins his daily work period;
- positions where the continuous driving time of the driver reaches a multiple of three hours;
- places and positions where the driver and/or the co-driver ends his daily work period.

110) When the position of the vehicle is not available from the GNSS receiver at these times, the recording equipment shall use the latest available position, and the related date and time.

111) Together with each place or position, the recording equipment shall record and store in its data memory:

- the (co-)driver card number and card issuing Member State,
- the card generation,
- the date and time of the entry,
- the type of entry (begin, end or 3 hours continuous driving time),
- the related GNSS accuracy, date and time if applicable;
- the vehicle odometer value.

112) The data memory shall be able to hold places and positions where daily work periods begin, end and/or where 3 hours continuous driving time is reached for at least 365 days.

113) When storage capacity is exhausted, new data shall replace oldest data.

#### 2.12.6 Odometer data

114) The recording equipment shall record in its data memory the vehicle odometer value and the corresponding date at midnight every calendar day.

115) The data memory shall be able to store midnight odometer values for at least 365 calendar days.

116) When storage capacity is exhausted, new data shall replace oldest data.

#### 2.12.7 Detailed speed data

117) The recording equipment shall record and store in its data memory the instantaneous speed of the vehicle and the corresponding date and time at every second of at least the last 24 hours that the vehicle has been driven.

#### 2.12.8 Events data

For the purpose of this subparagraph, time shall be recorded with a resolution of 1 second.

118) The recording equipment shall record and store in its data memory the following data for each event detected according to the following storage rules:

119)

<i>Event</i>	<i>Storage rules</i>	<i>Data to be recorded per event</i>
Insertion of a non-valid card	- the 10 most recent events.	- date and time of event, - card(s) type, number, issuing Member State and generation of the card creating the event. - number of similar events that day
Card conflict	- the 10 most recent events.	- date and time of beginning of event, - date and time of end of event, - card(s) type, number, issuing Member State and generation of the two cards creating the conflict.
Driving without an appropriate card	- the longest event for each of the 10 last days of occurrence, - the 5 longest events over the last 365 days.	- date and time of beginning of event, - date and time of end of event, - card(s) type, number, issuing Member State and generation of any card inserted at beginning and/or end of the event, - number of similar events that day.
Card insertion while driving	- the last event for each of the 10 last days of occurrence,	- date and time of the event, - card(s) type, number, issuing Member State and generation, - number of similar events that day

<i>Event</i>	<i>Storage rules</i>	<i>Data to be recorded per event</i>
Last card session not correctly closed	- the 10 most recent events.	- date and time of card insertion, - card(s) type, number, issuing Member State and generation, - last session data as read from the card: - date and time of card insertion, - VRN, Member State of registration and VU generation.
Over speeding (1)	- the most serious event for each of the 10 last days of occurrence (i.e. the one with the highest average speed), - the 5 most serious events over the last 365 days. - the first event having occurred after the last calibration	- date and time of beginning of event, - date and time of end of event, - maximum speed measured during the event, - arithmetic average speed measured during the event, - card type, number, issuing Member State and generation of the driver card (if applicable), - number of similar events that day.
Power supply interruption (2)	- the longest event for each of the 10 last days of occurrence, - the 5 longest events over the last 365 days.	- date and time of beginning of event, - date and time of end of event, - card(s) type, number, issuing Member State and generation of any card inserted at beginning and/or end of the event, - number of similar events that day.

<i>Event</i>	<i>Storage rules</i>	<i>Data to be recorded per event</i>
Communication error with the remote communication facility	<ul style="list-style-type: none"> <li>- the longest event for each of the 10 last days of occurrence,</li> <li>- the 5 longest events over the last 365 days.</li> </ul>	<ul style="list-style-type: none"> <li>- date and time of beginning of event,</li> <li>- date and time of end of event,</li> <li>- card(s) type, number, issuing Member State and generation of any card inserted at beginning and/or end of the event,</li> <li>- number of similar events that day.</li> </ul>
Absence of position information from GNSS receiver	<ul style="list-style-type: none"> <li>- the longest event for each of the 10 last days of occurrence,</li> <li>- the 5 longest events over the last 365 days.</li> </ul>	<ul style="list-style-type: none"> <li>- date and time of beginning of event,</li> <li>- date and time of end of event,</li> <li>- card(s) type, number, issuing Member State and generation of any card inserted at beginning and/or end of the event,</li> <li>- number of similar events that day.</li> </ul>
Motion data error	<ul style="list-style-type: none"> <li>- the longest event for each of the 10 last days of occurrence,</li> <li>- the 5 longest events over the last 365 days.</li> </ul>	<ul style="list-style-type: none"> <li>- date and time of beginning of event,</li> <li>- date and time of end of event,</li> <li>- card(s) type, number, issuing Member State and generation of any card inserted at beginning and/or end of the event,</li> <li>- number of similar events that day.</li> </ul>



<i>Event</i>	<i>Storage rules</i>	<i>Data to be recorded per event</i>
Vehicle motion conflict	<ul style="list-style-type: none"> <li>- the longest event for each of the 10 last days of occurrence,</li> <li>- the 5 longest events over the last 365 days.</li> </ul>	<ul style="list-style-type: none"> <li>- date and time of beginning of event,</li> <li>- date and time of end of event,</li> <li>- card(s) type, number, issuing Member State and generation of any card inserted at beginning and/or end of the event,</li> <li>- number of similar events that day.</li> </ul>
Security breach attempt	<ul style="list-style-type: none"> <li>- the 10 most recent events per type of event.</li> </ul>	<ul style="list-style-type: none"> <li>- date and time of beginning of event,</li> <li>- date and time of end of event (if relevant),</li> <li>- card(s) type, number, issuing Member State and generation of any card inserted at beginning and/or end of the event,</li> <li>- type of event.</li> </ul>
Time conflict	<ul style="list-style-type: none"> <li>- the longest event for each of the 10 last days of occurrence,</li> <li>- the 5 longest events over the last 365 days.</li> </ul>	<ul style="list-style-type: none"> <li>- recording equipment date and time</li> <li>- GNSS date and time,</li> <li>- card(s) type, number, issuing Member State and generation of any card inserted at beginning and/or end of the event,</li> <li>- number of similar events that day.</li> </ul>

(1) The recording equipment shall also record and store in its data memory:

- the date and time of the last OVER SPEEDING CONTROL,
- the date and time of the first over speeding following this OVER SPEEDING CONTROL,
- the number of over speeding events since the last OVER SPEEDING CONTROL.

(2) These data may be recorded at power supply reconnection only, times may be known with an accuracy to the minute.

2.12.9 Faults data

For the purpose of this subparagraph, time shall be recorded with a resolution of 1 second.

120) The recording equipment shall attempt to record and store in its data memory the following data for each fault detected according to the following storage rules:

<i>Fault</i>	<i>Storage rules</i>	<i>Data to be recorded per fault</i>
Card fault	- the 10 most recent driver card faults.	- date and time of beginning of fault, - date and time of end of fault, - card(s) type, number, issuing Member State and generation.
Recording equipment faults	- the 10 most recent faults for each type of fault, - the first fault after the last calibration.	- date and time of beginning of fault, - date and time of end of fault, - type of fault, - card(s) type, number and issuing Member State and generation of any card inserted at beginning and/or end of the fault.

2.12.10 Calibration data

121) The recording equipment shall record and store in its data memory data relevant to:

- known calibration parameters at the moment of activation,
- its very first calibration following its activation,
- its first calibration in the current vehicle (as identified by its VIN),
- the 20 most recent calibrations (if several calibrations happen within one calendar day, only the first and the last one of the day shall be stored).

122) The following data shall be recorded for each of these calibrations:

- purpose of calibration (activation, first installation, installation, periodic inspection),
- workshop name and address,
- workshop card number, card issuing Member State and card expiry date,
- vehicle identification,
- parameters updated or confirmed: w, k, l, tyre size, speed limiting device setting, odometer (old and new values), date and time (old and new values),
- the types and identifiers of all the seals in place.

- 123) In addition, the recording equipment shall record and store in its data memory its ability to use first generation tachograph cards (still activated or not).
- 124) The motion sensor shall record and store in its memory the following motion sensor installation data:
- first pairing with a VU (date, time, VU approval number, VU serial number),
  - last pairing with a VU (date, time, VU approval number, VU serial number).
- 125) The external GNSS facility shall record and store in its memory the following external GNSS facility installation data:
- first coupling with a VU (date, time, VU approval number, VU serial number),
  - last coupling with a VU (date, time, VU approval number, VU serial number).

#### 2.12.11 Time adjustment data

- 126) The recording equipment shall record and store in its data memory data relevant to time adjustments performed in calibration mode outside the frame of a regular calibration (def. f):
- the most recent time adjustment,
  - the 5 largest time adjustments.
- 127) The following data shall be recorded for each of these time adjustments:
- date and time, old value,
  - date and time, new value,
  - workshop name and address,
  - workshop card number, card issuing Member State, card generation and card expiry date.

#### 2.12.12 Control activity data

- 128) The recording equipment shall record and store in its data memory the following data relevant to the 20 most recent control activities:
- date and time of the control,
  - control card number, card issuing Member State and card generation,
  - type of the control (displaying and/or printing and/or VU downloading and/or card downloading and/or roadside calibration checking).
- 129) In case of downloading, the dates of the oldest and of the most recent days downloaded shall also be recorded.

#### 2.12.13 Company locks data

- 130) The recording equipment shall record and store in its data memory the following data relevant to the 255 most recent company locks:
- lock-in date and time,
  - lock-out date and time,
  - company card number, card issuing Member State and card generation,
  - company name and address.

Data previously locked by a lock removed from memory due to the limit above, shall be treated as not locked.

#### 2.12.14 Download activity data

- 131) The recording equipment shall record and store in its data memory the following data relevant to the last data memory downloading to external media while in company or in calibration mode:
- date and time of downloading,
  - company or workshop card number, card issuing Member State and card generation,
  - company or workshop name.

#### 2.12.15 Specific conditions data

- 132) The recording equipment shall record in its data memory the following data relevant to specific conditions:
- date and time of the entry,
  - type of specific condition.

- 133) The data memory shall be able to hold specific conditions data for at least 365 days (with the assumption that on average, 1 condition is opened and closed per day). When storage capacity is exhausted, new data shall replace oldest data.

#### 2.12.16 Tachograph card data

- 134) The recording equipment shall be able to store the following data related to the different tachograph cards in which had been used in the VU:
- the tachograph card number and its serial number,
  - the manufacturer of the tachograph card,
  - the tachograph card type,
  - the tachograph card version.

- 135) The recording equipment shall be able to store at least 88 such records.

### 2.13 Reading from tachograph cards

- 136) The recording equipment shall be able to read from first and second generation tachograph cards, where applicable, the necessary data:
- to identify the card type, the card holder, the previously used vehicle, the date and time of the last card withdrawal and the activity selected at that time,
  - to check that last card session was correctly closed,
  - to compute the driver’s continuous driving time, cumulative break time and cumulated driving times for the previous and the current week,
  - to print requested printouts related to data recorded on a driver card,
  - to download a driver card to external media.

This requirement only applies to first generation tachograph cards if their use has not been suppressed by a workshop.

- 137) In case of a reading error, the recording equipment shall try again, three times maximum, the same read command, and then if still unsuccessful, declare the card faulty and non-valid.

### 2.14 Recording and storing on tachograph cards

#### 2.14.1 Recording and storing in first generation tachograph cards

- 138) Provided first generation tachograph cards use has not been suppressed by a workshop, the recording equipment shall record and store data exactly in the same way as a first generation recording equipment would do.
- 139) The recording equipment shall set the “card session data” in the driver or workshop card right after the card insertion.
- 140) The recording equipment shall update data stored on valid driver, workshop, company and/or control cards with all necessary data relevant to the period while the card is inserted and relevant to the card holder. Data stored on these cards are specified in Chapter 4.
- 141) The recording equipment shall update driver activity and places data (as specified in 4.5.3.1.9 and 4.5.3.1.11), stored on valid driver and/or workshop cards, with activity and places data manually entered by the cardholder.
- 142) All events not defined for the first generation recording equipment, shall not be stored on the driver and workshop cards.
- 143) Tachograph cards data update shall be such that, when needed and taking into account card actual storage capacity, most recent data replace oldest data.

144) In the case of a writing error, the recording equipment shall try again, three times maximum, the same write command and then if still unsuccessful, declare the card faulty and non-valid.

145) Before releasing a driver card and after all relevant data have been stored on the card, the recording equipment shall reset the “card session data”.

#### 2.14.2 Recording and storing in second generation tachograph cards

146) Second generation tachograph cards shall contain 2 different card applications, the first of which shall be exactly the same as the TACHO application of first generation tachograph cards, and the second the “TACHO\_G2” application, as specified in Chapter 4 and Appendix 2.

147) The recording equipment shall set the “card session data” in the driver or workshop card right after the card insertion.

148) The recording equipment shall update data stored on the 2 card applications of valid driver, workshop, company and/or control cards with all necessary data relevant to the period while the card is inserted and relevant to the card holder. Data stored on these cards are specified in Chapter 4.

149) The recording equipment shall update driver activity places and positions data (as specified in 4.5.3.1.9, 4.5.3.1.11, 4.5.3.2.9 and 4.5.3.2.11), stored on valid driver and/or workshop cards, with activity and places data manually entered by the cardholder.

150) Tachograph cards data update shall be such that, when needed and taking into account card actual storage capacity, most recent data replace oldest data.

151) In the case of a writing error, the recording equipment shall try again, three times maximum, the same write command and then if still unsuccessful, declare the card faulty and non-valid.

152) Before releasing a driver card and after all relevant data have been stored on the 2 card applications of the card, the recording equipment shall reset the “card session data”.

#### 2.15 Displaying

153) The display shall include at least 20 characters.

154) The minimum character size shall be 5 mm high and 3.5 mm wide.

155) The display shall support the characters specified in Appendix 1 Chapter 4 ‘Character sets’. The display may use simplified glyphs (e.g. accented characters may be displayed without accent, or lower case letters may be shown as upper case letters).

156) The display shall be provided with adequate non-dazzling lighting.

157) Indications shall be visible from outside the recording equipment.

158) The recording equipment shall be able to display:

- default data,
- data related to warnings,
- data related to menu access,
- other data requested by a user.

Additional information may be displayed by the recording equipment, provided that it is clearly distinguishable from information required above.

159) The display of the recording equipment shall use the pictograms or pictograms combinations listed in Appendix 3. Additional pictograms or pictograms combinations may also be provided by the display, if clearly distinguishable from the aforementioned pictograms or pictograms combinations.

160) The display shall always be ON when the vehicle is moving.

161) The recording equipment may include a manual or automatic feature to turn the display OFF when the vehicle is not moving.

Displaying format is specified in Appendix 5.

#### 2.15.1 Default display

162) When no other information needs to be displayed, the recording equipment shall display, by default, the following:

- the local time (as a result of UTC time + offset as set by the driver),
- the mode of operation,
- the current activity of the driver and the current activity of the co-driver,
- information related to the driver:
  - if his current activity is DRIVING, his current continuous driving time and his current cumulative break time,
  - if his current activity is not DRIVING, the current duration of this activity (since it was selected) and his current cumulative break time.

- 163) Display of data related to each driver shall be clear, plain and unambiguous. In the case where the information related to the driver and the co-driver cannot be displayed at the same time, the recording equipment shall display by default the information related to the driver and shall allow the user to display the information related to the co-driver.
- 164) In the case where the display width does not allow displaying by default the mode of operation, the recording equipment shall briefly display the new mode of operation when it changes.
- 165) The recording equipment shall briefly display the card holder name at card insertion.
- 166) When an “OUT OF SCOPE” or FERRY/TRAIN condition is opened, then the default display must show using the relevant pictogram that the particular condition is opened (it is acceptable that the driver’s current activity may not be shown at the same time).

#### 2.15.2 Warning display

- 167) The recording equipment shall display warning information using primarily the pictograms of Appendix 3, completed where needed by additional numerically coded information. A literal description of the warning may also be added in the driver’s preferred language.

#### 2.15.3 Menu access

- 168) The recording equipment shall provide necessary commands through an appropriate menu structure.

#### 2.15.4 Other displays

- 169) It shall be possible to display selectively on request:
- the UTC date and time, and local time offset,
  - the content of any of the six printouts under the same formats as the printouts themselves,
  - the continuous driving time and cumulative break time of the driver,
  - the continuous driving time and cumulative break time of the co-driver,
  - the cumulated driving time of the driver for the previous and the current week,
  - the cumulated driving time of the co-driver for the previous and the current week,

optional:

- the current duration of co-driver activity (since it was selected),
- the cumulated driving time of the driver for current week,
- the cumulated driving time of the co-driver for the current daily work period,
- the cumulated driving time of the driver for the current daily work period.



170) Printout content display shall be sequential, line by line. If the display width is less than 24 characters the user shall be provided with the complete information through an appropriate mean (several lines, scrolling, ...).

Printout lines devoted to hand-written information may be omitted for display.

#### 2.16 Printing

171) The recording equipment shall be able to print information from its data memory and/or from tachograph cards in accordance with the seven following printouts:

- driver activities from card daily printout,
- driver activities from Vehicle Unit daily printout,
- events and faults from card printout,
- events and faults from Vehicle Unit printout,
- technical data printout,
- over speeding printout.
- tachograph card data history for a given VU (see chapter 3.12.16)

The detailed format and content of these printouts are specified in Appendix 4.

Additional data may be provided at the end of the printouts.

Additional printouts may also be provided by the recording equipment, if clearly distinguishable from the seven aforementioned printouts.

172) The “driver activities from card daily printout” and “Events and faults from card printout” shall be available only when a driver card or a workshop card is inserted in the recording equipment. The recording equipment shall update data stored on the relevant card before starting printing.

173) In order to produce the “driver activities from card daily printout” or the “events and faults from card printout”, the recording equipment shall:

- either automatically select the driver card or the workshop card if one only of these cards is inserted,
- or provide a command to select the source card or select the card in the driver slot if two of these cards are inserted in the recording equipment.

174) The printer shall be able to print 24 characters per line.

175) The minimum character size shall be 2.1 mm high and 1.5 mm wide.

- 176) The printer shall support the characters specified in Appendix 1 Chapter 4 'Character sets'.
- 177) Printers shall be so designed as to produce these printouts with a degree of definition likely to avoid any ambiguity when they are read.
- 178) Printouts shall retain their dimensions and recordings under normal conditions of humidity (10-90%) and temperature.
- 179) The type approved paper used by the recording equipment shall bear the relevant type approval mark and an indication of the type(s) of recording equipment with which it may be used.
- 180) Printouts shall remain clearly legible and identifiable under normal conditions of storage, in terms of light intensity, humidity and temperature, for at least two years.
- 181) Printouts shall conform at least to the test specifications defined in Appendix 9.
- 182) It shall also be possible to add hand-written notes, such as the driver's signature, to these documents.
- 183) The recording equipment shall manage "paper out" events while printing by, once paper has been re-loaded, restarting printing from printout beginning or by continuing printing and providing an unambiguous reference to previously printed part.
- 2.17 Warnings
- 184) The recording equipment shall warn the driver when detecting any event and/or fault.
- 185) Warning of a power supply interruption event may be delayed until the power supply is reconnected.
- 186) The recording equipment shall warn the driver 15 minutes before and at the time of exceeding the maximum allowed continuous driving time.
- 187) Warnings shall be visual. Audible warnings may also be provided in addition to visual warnings.
- 188) Visual warnings shall be clearly recognisable by the user, shall be situated in the driver's field of vision and shall be clearly legible both by day and by night.
- 189) Visual warnings may be built into the recording equipment and/or remote from the recording equipment.

- 190) In the latter case it shall bear a “T” symbol.
- 191) Warnings shall have a duration of at least 30 seconds, unless acknowledged by the user by hitting one or more specific keys of the recording equipment. This first acknowledgement shall not erase warning cause display referred to in next paragraph.
- 192) Warning cause shall be displayed on the recording equipment and remain visible until acknowledged by the user using a specific key or command of the recording equipment.
- 193) Additional warnings may be provided, as long as they do not confuse drivers in relation to previously defined ones.

#### 2.18 Data downloading to external media

- 194) The recording equipment shall be able to download on request data from its data memory or from a driver card to external storage media via the calibration/downloading connector. The recording equipment shall update data stored on the relevant card before starting downloading.
- 195) In addition and as an optional feature, the recording equipment may, in any mode of operation, download data through any another means to a company authenticated through this channel. In such a case, company mode data access rights shall apply to this download.
- 196) Downloading shall not alter or delete any stored data.
- 197) The calibration/downloading connector electrical interface is specified in Appendix 6.
- 198) Downloading protocols are specified in Appendix 7.

#### 2.19 Remote communication for targeted roadside checks

- 199) When the ignition is on, the Vehicle Unit shall store every 60 seconds in the remote communication facility the most recent data necessary for the purpose of targeted roadside checks. Such data shall be encrypted and signed as specified in Appendix 11 and Appendix 14.
- 200) Data to be checked remotely shall be available to remote communication readers through wireless communication, as specified in Appendix 14.
- 201) Data necessary for the purpose of targeted roadside checks shall be related to :
- the latest security breach attempt,

- the longest power supply interruption,
- sensor fault,
- motion data error,
- vehicle motion conflict,
- driving without a valid card,
- card insertion while driving,
- time adjustment data,
- calibration data including the dates of the two latest stored calibration records,
- vehicle registration number,
- speed recorded by the tachograph.

#### 2.20 Output data to additional external devices

202) The recording equipment may also be equipped with standardised interfaces allowing the data recorded or produced by tachograph to be used in operational or calibration mode, by an external facility.

In Appendix 13, an optional ITS interface is specified and standardized. Other similar interfaces may co-exist, provided they fully comply with the requirements of Appendix 13 in term of minimum list of data, security and driver consent.

The following requirements apply to ITS data made available through that interface:

- these data are a set of selected existing data from the tachograph data dictionary (Appendix 1),
- a subset of these selected data are marked 'personal data',
- the subset of 'personal data' is only available if the verifiable consent of the driver, accepting his personal data can leave the vehicle network, is enabled,
- At any moment, the driver consent can be enabled or disabled through commands in the menu, provided the driver card is inserted,
- the set and subset of data will be broadcasted via Bluetooth wireless protocol in the radius of the vehicle cab, with a refresh rate of 1 minute,
- the pairing of the external device with the ITS interface will be protected by a dedicated and random PIN of at least 4 digits, recorded in and available through the display of each vehicle unit,
- in any circumstances, the presence of the ITS interface cannot disturb or affect the correct functioning and the security of the vehicle unit.

Other data may also be output in addition to the set of selected existing data, considered as the minimum list, provided they cannot be considered as personal data.

The recording equipment shall notify other external facilities about the consent of the driver.

When the ignition of the vehicle is ON, these data shall be permanently broadcasted.

203) The serial link interface as specified in Annex 1B to Regulation (EEC) N°. 3821/85, as last amended, can continue to equip tachographs for back compatibility. Anyhow, the driver consent is still required in case personal data are transmitted.

#### 2.21 Calibration

204) The calibration function shall allow:

- to automatically pair the motion sensor with the VU,
- to automatically couple the external GNSS facility with the VU if applicable,
- to digitally adapt the constant of the recording equipment (k) to the characteristic coefficient of the vehicle (w) ,
- to adjust the current time within the validity period of the inserted workshop card,
- to adjust the current odometer value,
- to update motion sensor identification data stored in the data memory,
- to update, if applicable, external GNSS facility identification data stored in the data memory,
- to update the types and identifiers of all the seals in place,
- to update or confirm other parameters known to the recording equipment: vehicle identification, w, l, tyre size and speed limiting device setting if applicable.

205) In addition, the calibration function shall allow to suppress the use of first generation tachograph cards in the recording equipment, provided the conditions specified in Appendix 15 are met.

206) Pairing the motion sensor to the VU shall consist, at least, in:

- updating motion sensor installation data held by the motion sensor (as needed),
- copying from the motion sensor to the VU data memory the necessary motion sensor identification data.

207) Coupling the external GNSS facility to the VU shall consist, at least, in:

- updating external GNSS facility installation data held by the external GNSS facility (as needed),

- copying from the external GNSS facility to the VU data memory the necessary external GNSS facility identification data including the serial number of the external GNSS facility,

The coupling shall be followed by the verification of the GNSS position information.

- 208) The calibration function shall be able to input necessary data through the calibration/downloading connector in accordance with the calibration protocol defined in Appendix 8. The calibration function may also input necessary data through other means.

#### 2.22 Roadside calibration checking

- 209) The roadside calibration checking function shall allow reading the motion sensor serial number (possibly embedded in the adaptor) and the external GNSS facility serial number (when applicable), connected to the vehicle unit, at the time of the request.

- 210) This reading shall at least be possible on the vehicle unit display through commands in the menus.

- 211) The roadside calibration checking function shall also allow controlling the selection of the I/O mode of the calibration I/O signal line specified in Appendix 6, via the K-line interface. This shall be done through the ECUAdjustmentSession, as specified in Appendix 8, section 7 Control of Test Pulses – Input output control functional unit.

#### 2.23 Time adjustment

- 212) The time adjustment function shall allow for automatically adjusting the current time. Two time sources are used in the recording equipment for time adjustment: 1) the internal VU clock, 2) the GNSS receiver.

- 213) The time setting of the VU internal clock shall be automatically re-adjusted at intervals of 12 hours maximum. When this delay has expired and the GNSS signal is not available, the time setting shall be done as soon as the VU can access a valid time provided by GNSS receiver, according to the vehicle ignition conditions. The time reference for the automatic time setting of the VU internal clock shall be derived from the GNSS receiver. A time conflict event shall be triggered if the current time deviates more than one (1) minute from the time information provided by the GNSS receiver.

- 214) The time adjustment function shall also allow for triggered adjustment of the current time, in calibration mode.

#### 2.24 Performance characteristics

- 215) The Vehicle Unit shall be fully operational in the temperature range -20°C to 70°C, the external GNSS facility in the temperature range -20°C to 70°C, and the motion sensor in the temperature range -40°C to 135°C. Data memory content shall be preserved at temperatures down to -40°C.

- 216) The tachograph shall be fully operational in the humidity range 10% to 90%.

- 217) The seals used in the smart tachograph shall withstand the same conditions than those applicable to the tachograph components to which they are affixed.
- 218) The recording equipment shall be protected against over-voltage, inversion of its power supply polarity, and short circuits.
- 219) Motion sensors shall either:
- react to a magnetic field disturbing vehicle motion detection. In such circumstances, the vehicle unit will record and store a sensor fault (requirement 88) or,
  - have a sensing element that is protected from, or immune to, magnetic fields.
- 220) The recording equipment and the external GNSS facility shall conform to international regulation UN ECE R10 and shall be protected against electrostatic discharges and transients.

#### 2.25 Materials

- 221) All the constituent parts of the recording equipment shall be made of materials of sufficient stability and mechanical strength and with stable electrical and magnetic characteristics.
- 222) For normal conditions of use, all the internal parts of the equipment shall be protected against damp and dust.
- 223) The Vehicle Unit and the external GNSS facility shall meet the protection grade IP 40 and the motion sensor shall meet the protection grade IP 64, as per standard IEC 60529:1989 including A1:1999 and A2:2013..
- 224) The recording equipment shall conform to applicable technical specifications related to ergonomic design.
- 225) The recording equipment shall be protected against accidental damage.

#### 2.26 Markings

- 226) If the recording equipment displays the vehicle odometer value and speed, the following details shall appear on its display:
- near the figure indicating the distance, the unit of measurement of distance, indicated by the abbreviation “km”,
  - near the figure showing the speed, the entry “km/h”.

The recording equipment may also be switched to display the speed in miles per hour, in which case the unit of measurement of speed shall be shown by the abbreviation “mph”. The recording equipment may also be switched to display the distance in miles, in which case the unit of measurement of distance shall be shown by the abbreviation “mi”.

- 227) A descriptive plaque shall be affixed to each separate component of the recording equipment and shall show the following details:

- name and address of the equipment manufacturer,
- manufacturer's part number and year of manufacture of the equipment,
- equipment serial number,
- approval mark for the equipment type.

228) When physical space is not sufficient to show all above mentioned details, the descriptive plaque shall show at least: the manufacturer's name or logo, and the equipment's part number.

### 3 Construction and functional requirements for tachograph cards

#### 3.1 Visible data

The front page shall contain:

229) the words "Driver card" or "Control card" or "Workshop card" or "Company card" printed in capital letters in the official language or languages of the Member State issuing the card, according to the type of the card.

230) the name of the Member State issuing the card (optional);

231) the distinguishing sign of the Member State issuing the card, printed in negative in a blue rectangle and encircled by 12 yellow stars. The distinguishing signs shall be as follows:

<i>B</i>	<i>Belgium</i>	<i>LV</i>	<i>Latvia</i>
<i>BG</i>	<i>Bulgaria</i>	<i>L</i>	<i>Luxembourg</i>
<i>CZ</i>	<i>Czech Republic</i>	<i>LT</i>	<i>Lithuania</i>
<i>CY</i>	<i>Cyprus</i>	<i>M</i>	<i>Malta</i>
DK	Denmark	NL	The Netherlands
D	Germany	A	Austria
EST	Estonia	PL	Poland
GR	Greece	P	Portugal
		RO	Romania
		SK	Slovakia
		SLO	Slovenia
E	Spain	FIN	Finland
F	France	S	Sweden
HR	Croatia		
H	Hungary		
IRL	Ireland	UK	The United Kingdom



<i>B</i>	<i>Belgium</i>	<i>LV</i>	<i>Latvia</i>
<i>BG</i>	<i>Bulgaria</i>	<i>L</i>	<i>Luxembourg</i>
<i>CZ</i>	<i>Czech Republic</i>	<i>LT</i>	<i>Lithuania</i>
<i>CY</i>	<i>Cyprus</i>	<i>M</i>	<i>Malta</i>
<b>I</b>	<b>Italy</b>		

232) information specific to the card issued, numbered as follows:

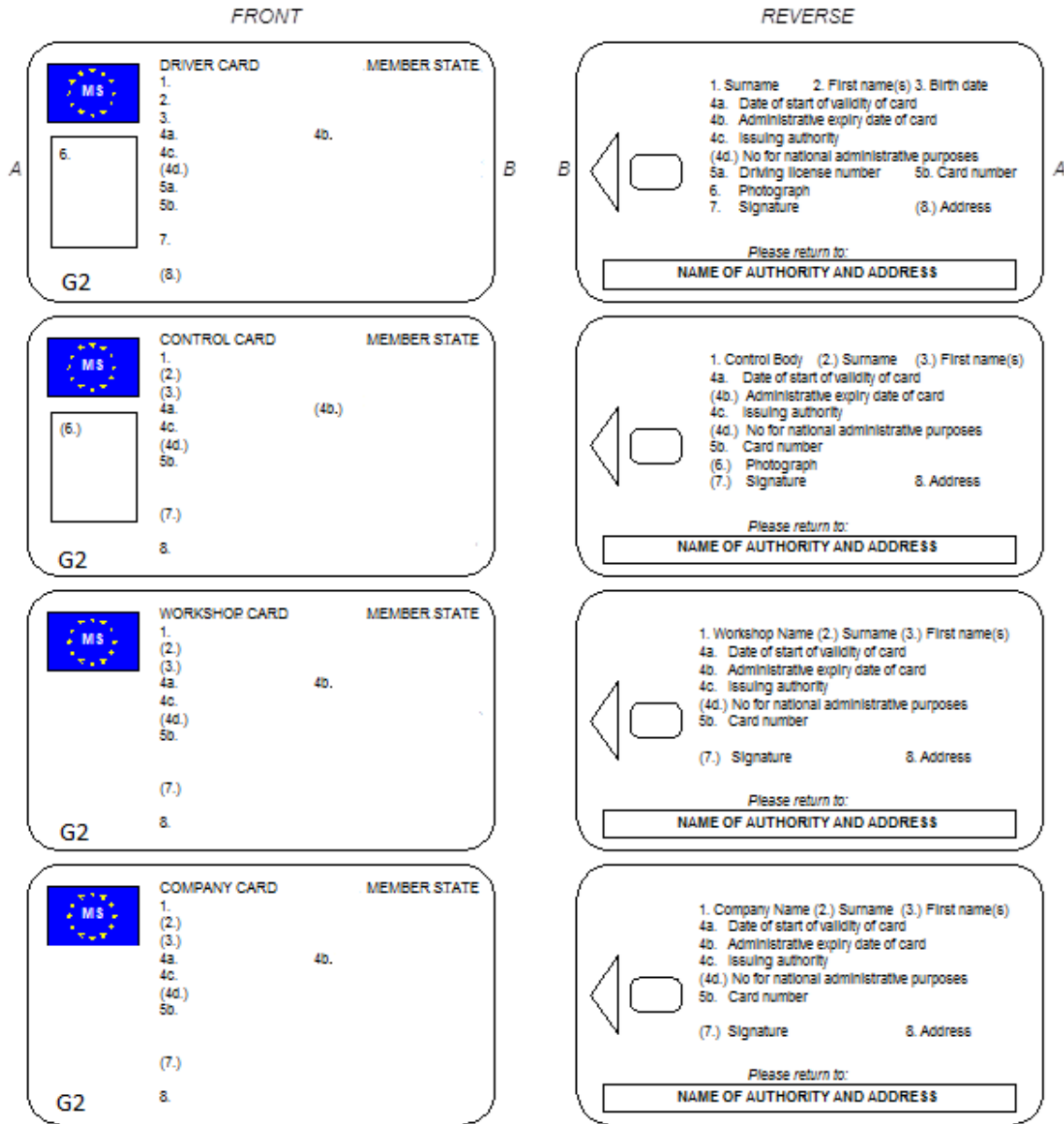
	<i>Driver card</i>	<i>Control Card</i>	<i>Company or Workshop card</i>
1.	surname of the driver	control body name	company or workshop name
2.	first name(s) of the driver	surname of the controller (if applicable)	surname of card holder (if applicable)
3.	birth date of the driver	first name(s) of the controller (if applicable)	first name(s) of card holder (if applicable)
4.a	card start of validity date		
4.b	card expiry date		
4.c	the name of the issuing authority (may be printed on reverse page)		
4.d	a different number from the one under heading 5, for administrative purposes (optional)		
5. a	Driving licence number  (at the date of issue of the driver card)	-	-
5. b	Card number		
6.	Photograph of the driver	photograph of the controller (optional)	photograph of the fitter (optional)-
7.	Signature of the holder (optional)		
8.	Normal place of residence, or postal address of the holder (optional).	Postal address of control body	postal address of company or workshop

233) dates shall be written using a “dd/mm/yyyy” or “dd.mm.yyyy” format (day, month, year).

The reverse page shall contain:

- 234) an explanation of the numbered items which appear on the front page of the card;
- 235) with the specific written agreement of the holder, information which is not related to the administration of the card may also be added, such addition will not alter in any way the use of the model as a tachograph card.
- 236) Tachograph cards shall be printed with the following background predominant colours:
- driver card: white,
  - control card: blue,
  - workshop card: red,
  - company card: yellow.
- 237) Tachograph cards shall bear at least the following features for protection of the card body against counterfeiting and tampering:
- a security design background with fine guilloche patterns and rainbow printing,
  - in the area of the photograph, the security design background and the photograph shall overlap,
  - at least one two-coloured microprint line.

COMMUNITY MODEL TACHOGRAPH CARDS



238) After consulting the Commission, Member States may add colours or markings, such as national symbols and security features, without prejudice to the other provisions of this Annex.

239) Temporary cards referred to in Article 26.4 of Regulation (EU) N°. 165/2014 shall comply with the provisions of this Annex.

### 3.2 Security

The system security aims at protecting integrity and authenticity of data exchanged between the cards and the recording equipment, protecting the integrity and authenticity of data downloaded from the cards, allowing certain write operations onto the cards to recording equipment only, decrypting certain data, ruling out any possibility of falsification of data stored in the cards, preventing tampering and detecting any attempt of that kind.

240) In order to achieve the system security, the tachograph cards shall meet the security requirements defined in Appendixes 10 and 11.

241) Tachograph cards shall be readable by other equipment such as personal computers.

### 3.3 Standards

242) Tachograph cards shall comply with the following standards:

- ISO/IEC 7810 Identification cards – Physical characteristics,
- ISO/IEC 7816 Identification cards - Integrated circuit cards:
  - Part 1: Physical characteristics,
  - Part 2: Dimensions and position of the contacts (ISO/IEC 7816-2:2007),
  - Part 3: Electrical interface and transmission protocols (ISO/IEC 7816-3:2006),
  - Part 4: Organisation, security and commands for interchange (ISO/IEC 7816-4:2013 + Cor 1:2014),
  - Part 6: Interindustry data elements for interchange (ISO/IEC 7816-6:2004 + Cor 1:2006),
  - Part 8: Commands for security operations (ISO/IEC 7816-8:2004).
- Tachograph cards shall be tested in accordance to ISO/IEC 10373-3 :2010 Identification cards – Test methods - Part 3: Integrated circuit cards with contacts and related interface devices.

### 3.4 Environmental and electrical specifications

243) Tachograph cards shall be capable of operating correctly in all the climatic conditions normally encountered in Community territory and at least in the temperature range -25°C to +70°C with occasional peaks of up to +85°C, “occasional” meaning not more than 4 hours each time and not over 100 times during the life time of the card.

244) Tachograph cards shall be capable of operating correctly in the humidity range 10% to 90%.

245) Tachograph cards shall be capable of operating correctly for a five-year period if used within the environmental and electrical specifications.

246) During operation, tachograph cards shall conform to ECE R10, related to electromagnetic compatibility, and shall be protected against electrostatic discharges.

### 3.5 Data storage

For the purpose of this paragraph,

- times are recorded with a resolution of one minute, unless otherwise specified,
- odometer values are recorded with a resolution of one kilometre,
- speeds are recorded with a resolution of 1 km/h,
- positions (latitudes and longitudes) are recorded in degrees and minutes with a resolution of 1/10 of minute.

The tachograph cards functions, commands and logical structures, fulfilling data storage requirements are specified in Appendix 2.

If not otherwise specified, data storage on tachograph cards shall be organized in such a way, that new data replaces stored oldest data in case the foreseen memory size for the particular records is exhausted.

247) This paragraph specifies minimum storage capacity for the various application data files. Tachograph cards shall be able to indicate to the recording equipment the actual storage capacity of these data files.

248) Any additional data that may be stored on tachograph cards, related to other applications possibly borne by the card, shall be stored in accordance with Directive 95/46/EC of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data<sup>8</sup> and with Directive 2002/58/EC of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector<sup>9</sup> and in compliance with Article 7 of Regulation (EU) N°. 165/2014.

249) Each Master File (MF) of any tachograph card shall contain up to five Elementary Files (EF) for card management, application and chip identifications, and two Dedicated Files (DF):

- DF Tachograph, which contains the application accessible to first generation vehicle units, which is also present in first generation tachograph cards,
- DF Tachograph\_G2, which contains the application only accessible to second generation vehicle units, which is only present in second generation tachograph cards.

The full details of the tachograph cards structure are specified in Appendix 2.

#### 3.5.1 Elementary files for identification and card management

#### 3.5.2 IC card identification

250) Tachograph cards shall be able to store the following smart card identification data:

- clock stop,

---

<sup>8</sup> OJ No L 281, 23/11/1995, p.31.

<sup>9</sup> OJ N° L 201, 31/07/2002, p.37

- card serial number (including manufacturing references),
- card type approval number,
- card personaliser identification (ID),
- embedder ID,
- IC identifier.

#### 3.5.2.1 Chip identification

251) Tachograph cards shall be able to store the following Integrated Circuit (IC) identification data:

- IC serial number,
- IC manufacturing references.

#### 3.5.2.2 DIR (only present in second generation tachograph cards)

252) Tachograph cards shall be able to store the application identification data objects specified in Appendix 2.

#### 3.5.2.3 ATR information (conditional, only present in second generation tachograph cards)

253) Tachograph cards shall be able to store the following extended length information data object:

- in the case the tachograph card supports extended length fields, the extended length information data object specified in Appendix 2.

#### 3.5.2.4 Extended length information (conditional, only present in second generation tachograph cards)

254) Tachograph cards shall be able to store the following extended length information data objects:

- in the case the tachograph card supports extended length fields, the extended length information data objects specified in Appendix 2.

### 3.5.3 Driver card

#### 3.5.3.1 Tachograph application (accessible to first and second generation vehicle units)

##### 3.5.3.1.1 Application identification

255) The driver card shall be able to store the following application identification data:

- tachograph application identification,
- type of tachograph card identification.

##### 3.5.3.1.2 Key and Certificates

256) The driver card shall be able to store a number of cryptographic keys and certificates, as specified in Appendix 11 part A.

##### 3.5.3.1.3 Card identification

257) The driver card shall be able to store the following card identification data:

- card number,
- issuing Member State, issuing authority name, issue date,
- card beginning of validity date, card expiry date.

#### 3.5.3.1.4 Card holder identification

258) The driver card shall be able to store the following card holder identification data:

- surname of the holder,
- first name(s) of the holder,
- date of birth,
- preferred language.

#### 3.5.3.1.5 Card download

259) The driver card shall be able to store the following data related to card download:

- date and time of last card download (for other purposes than control).

260) The driver card shall be able to hold one such record.

#### 3.5.3.1.6 Driving licence information

261) The driver card shall be able to store the following driving licence data:

- issuing Member State, issuing authority name,
- driving licence number (at the date of the issue of the card).

#### 3.5.3.1.7 Events data

For the purpose of this subparagraph, time shall be stored with a resolution of 1 second.

262) The driver card shall be able to store data related to the following events detected by the recording equipment while the card was inserted:

- Time overlap (where this card is the cause of the event),
- Card insertion while driving (where this card is the subject of the event),
- Last card session not correctly closed (where this card is the subject of the event),
- Power supply interruption,
- Motion data error,
- Security breach attempts.

263) The driver card shall be able to store the following data for these events:

- Event code,

- Date and time of beginning of the event (or of card insertion if the event was on-going at that time),
- Date and time of end of the event (or of card withdrawal if the event was on-going at that time),
- VRN and registering Member State of vehicle in which the event happened.

Note: For the “Time overlap” event:

- Date and time of beginning of the event shall correspond to the date and time of the card withdrawal from the previous vehicle,
- Date and time of end of the event shall correspond to the date and time of card insertion in current vehicle,
- Vehicle data shall correspond to the current vehicle raising the event.

Note: For the “Last card session not correctly closed” event:

- date and time of beginning of event shall correspond to the card insertion date and time of the session not correctly closed,
- date and time of end of event shall correspond to the card insertion date and time of the session during which the event was detected (current session),
- Vehicle data shall correspond to the vehicle in which the session was not correctly closed.

264) The driver card shall be able to store data for the six most recent events of each type (i.e. 36 events).

#### 3.5.3.1.8 Faults data

For the purpose of this subparagraph, time shall be recorded with a resolution of 1 second.

265) The driver card shall be able to store data related to the following faults detected by the recording equipment while the card was inserted:

- Card fault (where this card is the subject of the event),
- Recording equipment fault.

266) The driver card shall be able to store the following data for these faults:

- Fault code,
- Date and time of beginning of the fault (or of card insertion if the fault was on-going at that time),
- Date and time of end of the fault (or of card withdrawal if the fault was on-going at that time),
- VRN and registering Member State of vehicle in which the fault happened.

267) The driver card shall be able to store data for the twelve most recent faults of each type (i.e. 24 faults).

#### 3.5.3.1.9 Driver activity data

268) The driver card shall be able to store, for each calendar day where the card has been used or for which the driver has entered activities manually, the following data:



- the date,
- a daily presence counter (increased by one for each of these calendar days),
- the total distance travelled by the driver during this day,
- a driver status at 00:00,
- whenever the driver has changed of activity, and/or has changed of driving status, and/or has inserted or withdrawn his card:
  - the driving status (CREW, SINGLE),
  - the slot (DRIVER, CO-DRIVER),
  - the card status (INSERTED, NOT INSERTED),
  - the activity (DRIVING, AVAILABILITY, WORK, BREAK/REST),
  - the time of the change.

269) The driver card memory shall be able to hold driver activity data for at least 28 days (the average activity of a driver is defined as 93 activity changes per day).

270) The data listed under requirements 261, 264 and 266 shall be stored in a way allowing the retrieval of activities in the order of their occurrence, even in case of a time overlap situation.

#### 3.5.3.1.10 Vehicles used data

271) The driver card shall be able to store, for each calendar day where the card has been used, and for each period of use of a given vehicle that day (a period of use includes all consecutive insertion / withdrawal cycle of the card in the vehicle, as seen from the card point of view), the following data:

- date and time of first use of the vehicle (i.e. first card insertion for this period of use of the vehicle, or 00h00 if the period of use is on-going at that time),
- vehicle odometer value at that time,
- date and time of last use of the vehicle, (i.e. last card withdrawal for this period of use of the vehicle, or 23h59 if the period of use is on-going at that time),
- vehicle odometer value at that time,
- VRN and registering Member State of the vehicle.

272) The driver card shall be able to store at least 84 such records.

#### 3.5.3.1.11 Places where daily work periods start and/or end

273) The driver card shall be able to store the following data related to places where daily work periods begin and/or end, entered by the driver:

- the date and time of the entry (or the date/time related to the entry if the entry is made during the manual entry procedure),
- the type of entry (begin or end, condition of entry),

- the country and region entered,
- the vehicle odometer value.

274) The driver card memory shall be able to hold at least 42 pairs of such records.

#### 3.5.3.1.12 Card session data

275) The driver card shall be able to store data related to the vehicle which opened its current session:

- date and time the session was opened (i.e. card insertion) with a resolution of one second,
- VRN and registering Member State.

#### 3.5.3.1.13 Control activity data

276) The driver card shall be able to store the following data related to control activities:

- date and time of the control,
- control card number and card issuing Member State,
- type of the control (displaying and/or printing and/or VU downloading and/or card downloading (see note)),
- Period downloaded, in case of downloading,
- VRN and registering Member State of the vehicle in which the control happened.

Note: card downloading will only be recorded if performed through a recording equipment.

277) The driver card shall be able to hold one such record.

#### 3.5.3.1.14 Specific conditions data

278) The driver card shall be able to store the following data related to specific conditions entered while the card was inserted (whatever the slot):

- Date and time of the entry,
- Type of specific condition.

279) The driver card shall be able to store at least 56 such records.

### 3.5.3.2 Tachograph generation 2 application (not accessible to first generation vehicle unit)

#### 3.5.3.2.1 Application identification

280) The driver card shall be able to store the following application identification data:

- tachograph application identification,
- type of tachograph card identification.

#### 3.5.3.2.2 Keys and Certificates

281) The driver card shall be able to store a number of cryptographic keys and certificates, as specified in Appendix 11 part B.

#### 3.5.3.2.3 Card identification

282) The driver card shall be able to store the following card identification data:

- card number,
- issuing Member State, issuing authority name, issue date,
- card beginning of validity date, card expiry date.

#### 3.5.3.2.4 Card holder identification

283) The driver card shall be able to store the following card holder identification data:

- surname of the holder,
- first name(s) of the holder,
- date of birth,
- preferred language.

#### 3.5.3.2.5 Card download

284) The driver card shall be able to store the following data related to card download:

- date and time of last card download (for other purposes than control).

285) The driver card shall be able to hold one such record.

#### 3.5.3.2.6 Driving licence information

286) The driver card shall be able to store the following driving licence data:

- issuing Member State, issuing authority name,
- driving licence number (at the date of the issue of the card).

#### 3.5.3.2.7 Events data

For the purpose of this subparagraph, time shall be stored with a resolution of 1 second.

287) The driver card shall be able to store data related to the following events detected by the recording equipment while the card was inserted:

- Time overlap (where this card is the cause of the event),
- Card insertion while driving (where this card is the subject of the event),
- Last card session not correctly closed (where this card is the subject of the event),
- Power supply interruption,
- Communication error with the remote communication facility,
- Absence of position information from GNSS receiver event,
- Communication error with the external GNSS facility
- Motion data error,

- Vehicle motion conflict,
- Security breach attempts,
- Time conflict.

288) The driver card shall be able to store the following data for these events:

- Event code,
- Date and time of beginning of the event (or of card insertion if the event was on-going at that time),
- Date and time of end of the event (or of card withdrawal if the event was on-going at that time),
- VRN and registering Member State of vehicle in which the event happened.

Note: For the “Time overlap” event:

- Date and time of beginning of the event shall correspond to the date and time of the card withdrawal from the previous vehicle,
- Date and time of end of the event shall correspond to the date and time of card insertion in current vehicle,
- Vehicle data shall correspond to the current vehicle raising the event.

Note: For the “Last card session not correctly closed” event:

- date and time of beginning of event shall correspond to the card insertion date and time of the session not correctly closed,
- date and time of end of event shall correspond to the card insertion date and time of the session during which the event was detected (current session),
- Vehicle data shall correspond to the vehicle in which the session was not correctly closed.

289) The driver card shall be able to store data for the six most recent events of each type (i.e. 66 events).

#### 3.5.3.2.8 Faults data

For the purpose of this subparagraph, time shall be recorded with a resolution of 1 second.

290) The driver card shall be able to store data related to the following faults detected by the recording equipment while the card was inserted:

- Card fault (where this card is the subject of the event),
- Recording equipment fault.

291) The driver card shall be able to store the following data for these faults:

- Fault code,

- Date and time of beginning of the fault (or of card insertion if the fault was on-going at that time),
- Date and time of end of the fault (or of card withdrawal if the fault was on-going at that time),
- VRN and registering Member State of vehicle in which the fault happened.

292) The driver card shall be able to store data for the twelve most recent faults of each type (i.e. 24 faults).

#### 3.5.3.2.9 Driver activity data

293) The driver card shall be able to store, for each calendar day where the card has been used or for which the driver has entered activities manually, the following data:

- the date,
- a daily presence counter (increased by one for each of these calendar days),
- the total distance travelled by the driver during this day,
- a driver status at 00:00,
- whenever the driver has changed of activity, and/or has changed of driving status, and/or has inserted or withdrawn his card:
  - the driving status (CREW, SINGLE)
  - the slot (DRIVER, CO-DRIVER),
  - the card status (INSERTED, NOT INSERTED),
  - the activity (DRIVING, AVAILABILITY, WORK, BREAK/REST).
  - the time of the change,

294) The driver card memory shall be able to hold driver activity data for at least 28 days (the average activity of a driver is defined as 93 activity changes per day).

295) The data listed under requirements 286, 289 and 291 shall be stored in a way allowing the retrieval of activities in the order of their occurrence, even in case of a time overlap situation.

#### 3.5.3.2.10 Vehicles used data

296) The driver card shall be able to store, for each calendar day where the card has been used, and for each period of use of a given vehicle that day (a period of use includes all consecutive insertion / withdrawal cycle of the card in the vehicle, as seen from the card point of view), the following data:

- date and time of first use of the vehicle (i.e. first card insertion for this period of use of the vehicle, or 00h00 if the period of use is on-going at that time),
- vehicle odometer value at that first use time,
- date and time of last use of the vehicle, (i.e. last card withdrawal for this period of use of the vehicle, or 23h59 if the period of use is on-going at that time),
- vehicle odometer value at that last use time,
- VRN and registering Member State of the vehicle,

- VIN of the vehicle.

297) The driver card shall be able to store at least 84 such records.

#### 3.5.3.2.11 Places and positions where daily work periods start and/or end

298) The driver card shall be able to store the following data related to places where daily work periods begin and/or end, entered by the driver:

- the date and time of the entry (or the date/time related to the entry if the entry is made during the manual entry procedure),
- the type of entry (begin or end, condition of entry),
- the country and region entered,
- the vehicle odometer value,
- the vehicle position,
- the GNSS accuracy, date and time when the position was determined.

299) The driver card memory shall be able to hold at least 84 pairs of such records.

#### 3.5.3.2.12 Card session data

300) The driver card shall be able to store data related to the vehicle which opened its current session:

- date and time the session was opened (i.e. card insertion) with a resolution of one second,
- VRN and registering Member State.

#### 3.5.3.2.13 Control activity data

301) The driver card shall be able to store the following data related to control activities:

- date and time of the control,
- control card number and card issuing Member State,
- type of the control (displaying and/or printing and/or VU downloading and/or card downloading (see note)),
- Period downloaded, in case of downloading,
- VRN and registering Member State of the vehicle in which the control happened.

Note: security requirements imply that card downloading will only be recorded if performed through a recording equipment.

302) The driver card shall be able to hold one such record.

#### 3.5.3.2.14 Specific conditions data

303) The driver card shall be able to store the following data related to specific conditions entered while the card was inserted (whatever the slot):

- Date and time of the entry,

- Type of specific condition.

304) The driver card shall be able to store at least 56 such records.

#### 3.5.3.2.15 Vehicle units used data

305) The driver card shall be able to store the following data related to the different vehicle units in which the card was used:

- the date and time of the beginning of the period of use of the vehicle unit (i.e. first card insertion in the vehicle unit for the period),
- the manufacturer of the vehicle unit,
- the vehicle unit type,
- the vehicle unit software version number.

306) The driver card shall be able to store at least 84 such records.

#### 3.5.3.2.16 Three hours continuous driving places data

307) The driver card shall be able to store the following data related to the position of the vehicle where the continuous driving time of the driver reaches a multiple of three hours:

- the date and time when the continuous driving time of the card holder reaches a multiple of three hours,
- the position of the vehicle.
- the GNSS accuracy, date and time when the position was determined.

308) The driver card shall be able to store at least 252 such records.

### 3.5.4 Workshop card

#### 3.5.4.1 Tachograph application (accessible to first and second generation vehicle units)

##### 3.5.4.1.1 Application identification

309) The workshop card shall be able to store the following application identification data:

- tachograph application identification,
- type of tachograph card identification.

##### 3.5.4.1.2 Keys and Certificates

310) The workshop card shall be able to store a number of cryptographic keys and certificates, as specified in Appendix 11 part A.

311) The workshop card shall be able to store a Personal Identification Number (PIN code).

3.5.4.1.3 Card identification

312) The workshop card shall be able to store the following card identification data:

- card number,
- issuing Member State, issuing authority name, issue date,
- card beginning of validity date, card expiry date.

3.5.4.1.4 Card holder identification

313) The workshop card shall be able to store the following card holder identification data:

- workshop name,
- workshop address,
- surname of the holder,
- first name(s) of the holder,
- preferred language.

3.5.4.1.5 Card download

314) The workshop card shall be able to store a card download data record in the same manner as a driver card.

3.5.4.1.6 Calibration and time adjustment data

315) The workshop card shall be able to hold records of calibrations and/or time adjustments performed while the card is inserted in a recording equipment.

316) Each calibration record shall be able to hold the following data:

- Purpose of calibration (activation, first installation, installation, periodic inspection,),
- Vehicle identification,
- Parameters updated or confirmed (w, k, l, tyre size, speed limiting device setting, odometer (new and old values), date and time (new and old values)),
- Recording equipment identification (VU part number, VU serial number, motion sensor serial number).

317) The workshop card shall be able to store at least 88 such records.

318) The workshop card shall hold a counter indicating the total number of calibrations performed with the card.

319) The workshop card shall hold a counter indicating the number of calibrations performed since its last download.

3.5.4.1.7 Events and faults data

320) The workshop card shall be able to store events and faults data records in the same manner as a driver card.



321) The workshop card shall be able to store data for the three most recent events of each type (i.e. 18 events) and the six most recent faults of each type (i.e. 12 faults).

3.5.4.1.8 Driver activity data

322) The workshop card shall be able to store driver activity data in the same manner as a driver card.

323) The workshop card shall be able to hold driver activity data for at least 1 day of average driver activity.

3.5.4.1.9 Vehicles used data

324) The workshop card shall be able to store vehicles used data records in the same manner as a driver card.

325) The workshop card shall be able to store at least 4 such records.

3.5.4.1.10 Daily work periods start and/or end data

326) The workshop card shall be able to store daily works period start and/or end data records in the same manner as a driver card.

327) The workshop card shall be able to hold at least 3 pairs of such records.

3.5.4.1.11 Card session data

328) The workshop card shall be able to store a card session data record in the same manner as a driver card.

3.5.4.1.12 Control activity data

329) The workshop card shall be able to store a control activity data record in the same manner as a driver card.

3.5.4.1.13 Specific conditions data

330) The workshop card shall be able to store data relevant to specific conditions in the same manner as the driver card.

331) The workshop card shall be able to store at least 2 such records.

3.5.4.2 Tachograph Generation 2 application (not accessible to first generation vehicle unit)

3.5.4.2.1 Application identification

332) The workshop card shall be able to store the following application identification data:

- tachograph application identification,
- type of tachograph card identification.

3.5.4.2.2 Keys and Certificates

333) The workshop card shall be able to store a number of cryptographic keys and certificates, as specified in Appendix 11 part B.

334) The workshop card shall be able to store a Personal Identification Number (PIN code).

#### 3.5.4.2.3 Card identification

335) The workshop card shall be able to store the following card identification data:

- card number,
- issuing Member State, issuing authority name, issue date,
- card beginning of validity date, card expiry date.

#### 3.5.4.2.4 Card holder identification

336) The workshop card shall be able to store the following card holder identification data:

- workshop name,
- workshop address,
- surname of the holder,
- first name(s) of the holder,
- preferred language.

#### 3.5.4.2.5 Card download

337) The workshop card shall be able to store a card download data record in the same manner as a driver card.

#### 3.5.4.2.6 Calibration and time adjustment data

338) The workshop card shall be able to hold records of calibrations and/or time adjustments performed while the card is inserted in a recording equipment.

339) Each calibration record shall be able to hold the following data:

- purpose of calibration (activation, first installation, installation, periodic inspection),
- vehicle identification,
- parameters updated or confirmed (w, k, l, tyre size, speed limiting device setting, odometer (new and old values), date and time (new and old values),
- recording equipment identification (VU part number, VU serial number, motion sensor serial number, remote communication facility serial number and external GNSS facility serial number, if applicable),
- seal type and identifier of all seals in place,
- ability of the VU to use first generation tachograph cards (enabled or not).

340) The workshop card shall be able to store at least 88 such records.

341) The workshop card shall hold a counter indicating the total number of calibrations performed with the card.

342) The workshop card shall hold a counter indicating the number of calibrations performed since its last download.

3.5.4.2.7 Events and faults data

343) The workshop card shall be able to store events and faults data records in the same manner as a driver card.

344) The workshop card shall be able to store data for the three most recent events of each type (i.e. 33 events) and the six most recent faults of each type (i.e. 12 faults).

3.5.4.2.8 Driver activity data

345) The workshop card shall be able to store driver activity data in the same manner as a driver card.

346) The workshop card shall be able to hold driver activity data for at least 1 day of average driver activity.

3.5.4.2.9 Vehicles used data

347) The workshop card shall be able to store vehicles used data records in the same manner as a driver card.

348) The workshop card shall be able to store at least 4 such records.

3.5.4.2.10 Daily work periods start and/or end data

349) The workshop card shall be able to store daily works period start and/or end data records in the same manner as a driver card.

350) The workshop card shall be able to hold at least 3 pairs of such records.

3.5.4.2.11 Card session data

351) The workshop card shall be able to store a card session data record in the same manner as a driver card.

3.5.4.2.12 Control activity data

352) The workshop card shall be able to store a control activity data record in the same manner as a driver card.

3.5.4.2.13 Vehicle units used data

353) The workshop card shall be able to store the following data related to the different vehicle units in which the card was used:

- the date and time of the beginning of the period of use of the vehicle unit (i.e. first card insertion in the vehicle unit for the period),
- the manufacturer of the vehicle unit,
- the vehicle unit type,
- the vehicle unit software version number.

354) The workshop card shall be able to store at least 4 such records.

3.5.4.2.14 Three hours continuous driving places data

355) The workshop card shall be able to store the following data related to the position of the vehicle where the continuous driving time of the driver reaches a multiple of three hours:

- the date and time when the continuous driving time of the card holder reaches a multiple of three hours,
- the position of the vehicle,
- the GNSS accuracy, date and time when the position was determined.

356) The workshop card shall be able to store at least 18 such records.

3.5.4.2.15 Specific conditions data

357) The workshop card shall be able to store data relevant to specific conditions in the same manner as the driver card.

358) The workshop card shall be able to store at least 2 such records.

3.5.5 Control card

3.5.5.1 Tachograph application (accessible to first and second generation vehicle units)

3.5.5.1.1 Application identification

359) The control card shall be able to store the following application identification data:

- tachograph application identification,
- type of tachograph card identification.

3.5.5.1.2 Keys and Certificates

360) The control card shall be able to store a number of cryptographic keys and certificates, as specified in Appendix 11 part A.

3.5.5.1.3 Card identification

361) The control card shall be able to store the following card identification data:

- card number,
- issuing Member State, issuing authority name, issue date,
- card beginning of validity date, card expiry date (if any).

3.5.5.1.4 Card holder identification

362) The control card shall be able to store the following card holder identification data:

- control body name,
- control body address,
- surname of the holder,
- first name(s) of the holder,

- preferred language.

#### 3.5.5.1.5 Control activity data

363) The control card shall be able to store the following control activity data:

- date and time of the control,
- type of the control (displaying and/or printing and/or VU downloading and/or card downloading and/or roadside calibration checking),
- period downloaded (if any),
- VRN and Member State registering authority of the controlled vehicle,
- card number and card issuing Member State of the driver card controlled.

364) The control card shall be able to hold at least 230 such records.

#### 3.5.5.2 Tachograph G2 application (not accessible to first generation vehicle unit)

##### 3.5.5.2.1 Application identification

365) The control card shall be able to store the following application identification data:

- tachograph application identification,
- type of tachograph card identification.

##### 3.5.5.2.2 Keys and Certificates

366) The control card shall be able to store a number of cryptographic keys and certificates, as specified in Appendix 11 part B.

##### 3.5.5.2.3 Card identification

367) The control card shall be able to store the following card identification data:

- card number,
- issuing Member State, issuing authority name, issue date,
- card beginning of validity date, card expiry date (if any).

##### 3.5.5.2.4 Card holder identification

368) The control card shall be able to store the following card holder identification data:

- control body name,
- control body address,
- surname of the holder,
- first name(s) of the holder,
- preferred language.

#### 3.5.5.2.5 Control activity data

369) The control card shall be able to store the following control activity data:

- date and time of the control,
- type of the control (displaying and/or printing and/or VU downloading and/or card downloading and/or roadside calibration checking)
- period downloaded (if any),
- VRN and Member State registering authority of the controlled vehicle,
- card number and card issuing Member State of the driver card controlled.

370) The control card shall be able to hold at least 230 such records.

#### 3.5.6 Company card

##### 3.5.6.1 Tachograph application (accessible to first and second generation vehicle units)

###### 3.5.6.1.1 Application identification

371) The company card shall be able to store the following application identification data:

- tachograph application identification,
- type of tachograph card identification.

###### 3.5.6.1.2 Keys and Certificates

372) The company card shall be able to store a number of cryptographic keys and certificates, as specified in Appendix 11 part A.

###### 3.5.6.1.3 Card identification

373) The company card shall be able to store the following card identification data:

- card number,
- issuing Member State, issuing authority name, issue date,
- card beginning of validity date, card expiry date (if any).

###### 3.5.6.1.4 Card holder identification

374) The company card shall be able to store the following card holder identification data:

- company name,
- company address.

###### 3.5.6.1.5 Company activity data

375) The company card shall be able to store the following company activity data:

- date and time of the activity,
- type of the activity (VU locking in and/or out, and/or VU downloading and/or card downloading)
- period downloaded (if any),

- VRN and Member State registering authority of vehicle,
  - card number and card issuing Member State (in case of card downloading).
- 376) The company card shall be able to hold at least 230 such records.
- 3.5.6.2 Tachograph G2 application (not accessible to first generation vehicle unit)
- 3.5.6.2.1 Application identification
- 377) The company card shall be able to store the following application identification data:
- tachograph application identification,
  - type of tachograph card identification.
- 3.5.6.2.2 Keys and Certificates
- 378) The company card shall be able to store a number of cryptographic keys and certificates, as specified in Appendix 11 part B.
- 3.5.6.2.3 Card identification
- 379) The company card shall be able to store the following card identification data:
- card number,
  - issuing Member State, issuing authority name, issue date,
  - card beginning of validity date, card expiry date (if any).
- 3.5.6.2.4 Card holder identification
- 380) The company card shall be able to store the following card holder identification data:
- company name,
  - company address.
- 3.5.6.2.5 Company activity data
- 381) The company card shall be able to store the following company activity data:
- date and time of the activity,
  - type of the activity (VU locking in and/or out, and/or VU downloading and/or card downloading)
  - period downloaded (if any),
  - VRN and Member State registering authority of vehicle,
  - card number and card issuing Member State (in case of card downloading).
- 382) The company card shall be able to hold at least 230 such records.
- 4 Installation of recording equipment
- 4.1 Installation
- 383) New recording equipment shall be delivered non-activated to fitters or vehicle manufacturers, with all calibration parameters, as listed in Chapter 3.21, set to appropriate and valid default values. Where no particular value is

appropriate, literal parameters shall be set to strings of “?” and numeric parameters shall be set to “0”. Delivery of security relevant parts of the recording equipment can be restricted if required during security certification.

- 384) Before its activation, the recording equipment shall give access to the calibration function even if not in calibration mode.
- 385) Before its activation, the recording equipment shall neither record nor store data referred by points 3.12.3, 3.12.9 and 3.12.12 to 3.12.15 inclusive.
- 386) During installation, vehicle manufacturers shall pre-set all known parameters.
- 387) Vehicle manufacturers or fitters shall activate the installed recording equipment at the latest before the vehicle is used in scope of Regulation (EC) N°. 561/2006.
- 388) The activation of the recording equipment shall be triggered automatically by the first insertion of a valid workshop card in either of its card interface devices.
- 389) Specific pairing operations required between the motion sensor and the vehicle unit, if any, shall take place automatically before or during activation.
- 390) In a similar way, specific coupling operations between the external GNSS facility and the vehicle unit, if any, shall take place automatically before or during activation.
- 391) After its activation, the recording equipment shall fully enforce functions and data access rights.
- 392) After its activation, the recording equipment shall communicate to the remote communication facility the secured data necessary for the purpose of targeted roadside checks.
- 393) The recording and storing functions of the recording equipment shall be fully operational after its activation.
- 394) Installation shall be followed by a calibration. The first calibration may not necessarily include entry of the vehicle registration number (VRN), when it is not known by the approved workshop having to undertake this calibration. In these circumstances, it shall be possible, for the vehicle owner, and at this time only, to enter the VRN using his Company Card prior to using the vehicle in scope of Regulation (EC) N°. 561/2006 (e.g by using commands through an appropriate menu structure of the vehicle unit's man-machine interface.)<sup>10</sup>. Any update or confirmation of this entry shall only be possible using a Workshop Card.

---

<sup>10</sup> O.J. L. 102, 11.4.2006, p.1



- 395) The installation of an external GNSS facility requires the coupling with the vehicle unit and the subsequent verification of the GNSS position information.
- 396) The recording equipment must be positioned in the vehicle in such a way as to allow the driver to access the necessary functions from his seat.

#### 4.2 Installation plaque

- 397) After the recording equipment has been checked on installation, an installation plaque, engraved or printed in a permanent way, which is clearly visible and easily accessible shall be affixed onto the recording equipment. In cases where this is not possible, the plaque shall be affixed to the vehicle's "B" pillar so that it is clearly visible. For vehicles that do not have a "B" pillar, the installation plaque should be affixed to the doorframe on the driver's side of the vehicle and be clearly visible in all cases.

After every inspection by an approved fitter or workshop, a new plaque shall be affixed in place of the previous one.

- 398) The plaque shall bear at least the following details:

- name, address or trade name of the approved fitter or workshop,
- characteristic coefficient of the vehicle, in the form “ $w = \dots \text{ imp/km}$ ”,
- constant of the recording equipment, in the form “ $k = \dots \text{ imp/km}$ ”,
- effective circumference of the wheel tyres in the form “ $l = \dots \text{ mm}$ ”,
- tyre size,
- the date on which the characteristic coefficient of the vehicle and the effective circumference of the wheel tyres were measured,
- the vehicle identification number,
- the presence (or not) of an external GNSS facility,
- the serial number of the external GNSS facility,
- the serial number of the remote communication device,
- the serial number of all the seals in place,
- the part of the vehicle where the adaptor, if any, is installed,
- the part of the vehicle where the motion sensor is installed, if not connected to the gear-box or an adaptor is not being used,
- a description of the colour of the cable between the adaptor and that part of the vehicle providing its incoming impulses,
- the serial number of the embedded motion sensor of the adaptor.

- 399) For M1 and N1 vehicles only, and which are fitted with an adaptor in conformity with Regulation (EC) N°. 68/2009<sup>11</sup> as last amended and where it is not possible to include all the information necessary, as described in Requirement 396, a second, additional, plaque may be used. In such cases, this additional plaque shall contain at least the last four indents described in Requirement 396.

This second, additional plaque, if used, shall be affixed next to or beside the first primary plaque described in Requirement 396, and shall have the same protection level. Furthermore the secondary plaque shall also bear the name, address or trade name of the approved fitter or workshop that carried out the installation, and the date of installation.

#### 4.3 Sealing

- 400) The following parts shall be sealed:

- Any connection which, if disconnected, would cause undetectable alterations to be made or undetectable data loss (this may e.g. apply for the motion sensor fitting on the gearbox, the adaptor for M1/N1 vehicles, the external GNSS connection or the vehicle unit);
- The installation plaque, unless it is attached in such a way that it cannot be removed without the markings thereon being destroyed.

- 401) The seals mentioned above may be removed:

- In case of emergency,
- To install, to adjust or to repair a speed limitation device or any other device contributing to road safety, provided that the recording equipment continues to function reliably and correctly and is resealed by an approved fitter or workshop (in accordance with Chapter 6) immediately after fitting the speed limitation device or any other device contributing to road safety or within seven days in other cases.

- 402) On each occasion that these seals are broken a written statement giving the reasons for such action shall be prepared and made available to the competent authority.

- 403) Seals shall hold an identification number, allocated by its manufacturer. This number shall be unique and distinct from any other seal number allocated by any other seals manufacturer.

This unique identification number is defined as: MM NNNNNN by non-removable marking, with MM as unique manufacturer identification (database registration to be managed by EC) and NNNNNN seal alpha-numeric number, unique in the manufacturer domain.

- 404) The seals shall have a free space where approved fitters, workshops or vehicle manufacturers can add a special mark according the Article 22.3 of Regulation (EU) N° 165/2014.

This mark shall not cover the seal identification number.

---

<sup>11</sup> OJ L21, 24.1.2009, p.3

- 405) Seals manufacturers shall be registered in a dedicated database and shall make their identification seals numbers public through a procedure to be established by the European Commission.
- 406) Approved workshops and vehicle manufacturers shall, in the frame of Regulation (EU) N° 165/2014, only use seals from those of the seals manufacturers listed in the data base mentioned above.
- 407) Seal manufacturers and their distributors shall maintain full traceability records of the seals sold to be used in the frame of Regulation (EU) N° 165/2014 and shall be prepared to produce them to competent national authorities whenever need be.
- 408) Seals unique identification numbers shall be visible on the installation plaque.

## 5 Checks, inspections and repairs

Requirements on the circumstances in which seals may be removed, as referred to in Article 22.5 of Regulation (EU) N° 165/2014, are defined in Chapter 5.3 of this annex.

### 5.1 Approval of fitters, workshops and vehicle manufacturers

The Member States approve, regularly control and certify the bodies to carry out:

- installations,
- checks,
- inspections,
- repairs.

Workshop cards shall be issued only to fitters and/or workshops approved for the activation and/or the calibration of recording equipment in conformity with this annex and, unless duly justified:

- who are not eligible for a company card;
- and whose other professional activities do not present a potential compromise of the overall security of the system as required in Appendix 10.

### 5.2 Check of new or repaired instruments

- 409) Every individual device, whether new or repaired, shall be checked in respect of its proper operation and the accuracy of its reading and recordings, within the limits laid down in Chapter 3.2.1, 3.2.2, 3.2.3 and 3.3 by means of sealing in accordance with Chapter 5.3 and calibration.

### 5.3 Installation inspection

- 410) When being fitted to a vehicle, the whole installation (including the recording equipment) shall comply with the provisions relating to maximum tolerances laid down in Chapter 3.2.1, 3.2.2, 3.2.3 and 3.3.

### 5.4 Periodic inspections

- 411) Periodic inspections of the equipment fitted to the vehicles shall take place after any repair of the equipment, or after any alteration of the characteristic coefficient of the vehicle or of the effective circumference of the tyres, or after equipment UTC time is wrong by more than 20 minutes, or when the VRN has changed, and at least once within two years (24 months) of the last inspection.

- 412) These inspections shall include the following checks:

- that the recording equipment is working properly, including the data storage in tachograph cards function and the communication with remote communication readers,
- that compliance with the provisions of chapter 3.2.1 and 3.2.2 on the maximum tolerances on installation is ensured,
- that compliance with the provisions of chapter 3.2.3 and 3.3 is ensured,
- that the recording equipment carries the type approval mark,
- that the installation plaque, as defined by Requirement 396, and the descriptive plaque, as defined by Requirement 225, are affixed,
- the tyre size and the actual circumference of the tyres,
- that there are no manipulation devices attached to the equipment,
- that seals are correctly placed, in good state, that their identification numbers are valid (referenced seal manufacturer in the EC database) and that their identification numbers correspond to the installation plaque markings (see requirement 401 ).

- 413) If one of the events listed in Chapter 3.9 (Detection of Events and/or Faults) is found to have occurred since the last inspection and is considered by tachograph manufacturers and/or national authorities as potentially putting the security of the equipment at risk, the workshop shall:

- a. make a comparison between the motion sensor identification data of the motion sensor plugged into the gearbox with that of the paired motion sensor registered in the vehicle unit;
- b. check if the information recorded on the installation plaque matches with the information contained within the vehicle unit record;
- c. check if the motion sensor serial number and approval number, if printed on the body of the motion sensor, matches the information stored in the recording equipment data memory;

- d. compare identification data marked on the descriptive plaque of the external GNSS facility, if any, to the ones stored in the vehicle unit data memory;

414) Workshops shall keep traces in their inspection reports of any findings concerning broken seals or manipulations devices. These reports shall be kept by workshops for at least 2 years and made available to the Competent Authority whenever requested to do so.

415) These inspections shall include a calibration and a preventive replacement of the seals whose fitting is under the responsibility of workshops. .

#### 5.5 Measurement of errors

416) The measurement of errors on installation and during use shall be carried out under the following conditions, which are to be regarded as constituting standard test conditions:

- vehicle unladen, in normal running order,
- tyre pressures in accordance with the manufacturer's instructions,
- tyre wear, within the limits allowed by national law,
- vehicle movement:
  - the vehicle shall advance under its own engine power in a straight line on level ground and at a speed of  $50 \pm 5$  km/h. The measuring distance shall be at least 1000m.
  - provided that it is of comparable accuracy, alternative methods, such as a suitable test bench, may also be used for the test.

#### 5.6 Repairs

417) Workshops shall be able to download data from the recording equipment to give the data back to the appropriate transport company.

418) Approved workshops shall issue to transport companies a certificate of data un-downloadability where the malfunction of the recording equipment prevents previously recorded data to be downloaded, even after repair by this workshop. The workshops will keep a copy of each issued certificate for at least two years.

#### 6 Card Issuing

The card issuing processes set-up by the Member States shall conform to the following:

419) The card number of the first issue of a tachograph card to an applicant shall have a consecutive index (if applicable) and a replacement index and a renewal index set to "0".

420) The card numbers of all non-personal tachograph cards issued to a single control body or a single workshop or a single transport company shall have the same first 13 digits, and shall all have a different consecutive index.

- 421) A tachograph card issued in replacement of an existing tachograph card shall have the same card number than the replaced one except the replacement index which shall be raised by “1” (in the order 0, ..., 9, A, ..., Z).
- 422) A tachograph card issued in replacement of an existing tachograph card shall have the same card expiry date as the replaced one.
- 423) A tachograph card issued in renewal of an existing tachograph card shall have the same card number as the renewed one except the replacement index which shall be reset to “0” and the renewal index which shall be raised by “1” (in the order 0, ..., 9, A, ..., Z).
- 424) The exchange of an existing tachograph card, in order to modify administrative data, shall follow the rules of the renewal if within the same Member State, or the rules of a first issue if performed by another Member State.
- 425) The “card holder surname” for non-personal workshop or control cards shall be filled with workshop or control body name or with the fitter or control officer’s name would Member States so decide.
- 426) Member States shall exchange data electronically in order to ensure the uniqueness of driver cards that they issue in accordance with Article 31 of Regulation (EU) N° 165/2014.

## 7 Type approval of recording equipment and tachograph cards

### 7.1 General points

For the purpose of this chapter, the words “recording equipment” mean “recording equipment or its components”. No type approval is required for the cable(s) linking the motion sensor to the VU, the external GNSS facility to the VU or the remote communication facility to the VU. The paper, for use by the recording equipment, shall be considered as a component of the recording equipment.

Any manufacturer may ask for type approval of its component with any type of motion sensor, external GNSS facility and vice versa, provided each component complies with the requirements of this annex. Alternately, manufacturers may also ask for type approval of recording equipment.

- 427) Recording equipment shall be submitted for approval complete with any integrated additional devices.
- 428) Type approval of recording equipment and of tachograph cards shall include security related tests, functional tests and interoperability tests. Positive results to each of these tests are stated by an appropriate certificate.
- 429) Member States type approval authorities will not grant a type approval certificate as long as they do not hold:
- a security certificate,

- a functional certificate,
- and an interoperability certificate

for the recording equipment or the tachograph card, subject of the request for type approval.

- 430) Any modification in software or hardware of the equipment or in the nature of materials used for its manufacture shall, before being used, be notified to the authority which granted type-approval for the equipment. This authority shall confirm to the manufacturer the extension of the type approval, or may require an update or a confirmation of the relevant functional, security and/or interoperability certificates.
- 431) Procedures to upgrade in-situ recording equipment software shall be approved by the authority which granted type approval for the recording equipment. Software upgrade must not alter nor delete any driver activity data stored in the recording equipment. Software may be upgraded only under the responsibility of the equipment manufacturer.
- 432) Type approval of software modifications aimed to upgrade a previously type approved recording equipment may not be refused if such modifications only apply to functions not specified in this Annex. Software upgrade of a recording equipment may exclude the introduction of new character sets, if not technically feasible.

#### 7.2 Security certificate

- 433) The security certificate is delivered in accordance with the provisions of Appendix 10 of this Annex. Recording equipment components to be certified are vehicle unit, motion sensor, external GNSS facility and tachograph cards.
- 434) In the exceptional circumstance that the security certification authorities refuse to certify new equipment on the ground of obsolescence of the security mechanisms, type approval shall continue to be granted only in these specific and exceptional circumstances, and when no alternative solution, compliant with the Regulation, exists.
- 435) In this circumstance the Member State concerned shall, without delay, inform the European Commission, which shall, within twelve calendar months of the grant of the type approval, launch a procedure to ensure that the level of security is restored to its original levels.

#### 7.3 Functional certificate

- 436) Each candidate for type approval shall provide the Member State's type approval authority with all the material and documentation that the authority deems necessary.
- 437) Manufacturers shall provide the relevant samples of type approval candidate products and associated documentation required by laboratories appointed to perform functional tests, and within one month of the request being made. Any costs resulting from this request shall be borne by the requesting entity. Laboratories shall treat all commercially sensitive information in confidence.
- 438) A functional certificate shall be delivered to the manufacturer only after all functional tests specified in Appendix 9, at least, have been successfully passed.

- 439) The type approval authority delivers the functional certificate. This certificate shall indicate, in addition to the name of its beneficiary and the identification of the model, a detailed list of the tests performed and the results obtained.
- 440) The functional certificate of any recording equipment component shall also indicate the type approval numbers of the other type approved compatible recording equipment components tested for its certification.
- 441) The functional certificate of any recording equipment component shall also indicate the ISO or CEN standard against which the functional interface has been certified.

#### 7.4 Interoperability certificate

- 442) Interoperability tests are carried out by a single laboratory under the authority and responsibility of the European Commission.
- 443) The laboratory shall register interoperability test requests introduced by manufacturers in the chronological order of their arrival.
- 444) Requests will be officially registered only when the laboratory is in possession of:
- the entire set of material and documents necessary for such interoperability tests,
  - the corresponding security certificate,
  - the corresponding functional certificate,

The date of the registration of the request shall be notified to the manufacturer.

- 445) No interoperability tests shall be carried out by the laboratory, for recording equipment or tachograph cards that have not been granted a security certificate and a functional certificate, except in the exceptional circumstances described in Requirement 432.
- 446) Any manufacturer requesting interoperability tests shall commit to leave to the laboratory in charge of these tests the entire set of material and documents which he provided to carry out the tests.
- 447) The interoperability tests shall be carried out, in accordance with the provisions of Appendix 9 of this Annex, with respectively all the types of recording equipment or tachograph cards:
- for which type approval is still valid or,
  - for which type approval is pending and that have a valid interoperability certificate.



- 448) The interoperability tests shall cover all generations of recording equipment or tachograph cards still in use.
- 449) The interoperability certificate shall be delivered by the laboratory to the manufacturer only after all required interoperability tests have been successfully passed.
- 450) If the interoperability tests are not successful with one or more of the recording equipment or tachograph card(s), the interoperability certificate shall not be delivered, until the requesting manufacturer has realised the necessary modifications and has succeeded the interoperability tests. The laboratory shall identify the cause of the problem with the help of the manufacturers concerned by this interoperability fault and shall attempt to help the requesting manufacturer in finding a technical solution. In the case where the manufacturer has modified its product, it is the manufacturer's responsibility to ascertain from the relevant authorities that the security certificate and the functional certificates are still valid.
- 451) The interoperability certificate is valid for six months. It is revoked at the end of this period if the manufacturer has not received a corresponding type approval certificate. It is forwarded by the manufacturer to the type approval authority of the Member State who has delivered the functional certificate.

- 452) Any element that could be at the origin of an interoperability fault shall not be used for profit or to lead to a dominant position.

#### 7.5 Type approval certificate

- 453) The type approval authority of the Member State may deliver the type approval certificate as soon as it holds the three required certificates.
- 454) The type approval certificate of any recording equipment component shall also indicate the type approval numbers of the other type approved interoperable recording equipment.
- 455) The type approval certificate shall be copied by the type approval authority to the laboratory in charge of the interoperability tests at the time of deliverance to the manufacturer.
- 456) The laboratory competent for interoperability tests shall run a public web site on which will be updated the list of recording equipment or tachograph cards models:
- for which a request for interoperability tests have been registered,
  - having received an interoperability certificate (even provisional),
  - having received a type approval certificate.

#### 7.6 Exceptional procedure: first interoperability certificates for 2nd generation recording equipment and tachograph cards

- 457) Until four months after a first couple of 2<sup>nd</sup> generation recording equipment and 2<sup>nd</sup> generation tachograph cards (driver, workshop, control and company cards) have been certified to be interoperable, any interoperability certificate delivered (including the first ones), regarding requests registered during this period, shall be considered provisional.
- 458) If at the end of this period, all products concerned are mutually interoperable, all corresponding interoperability certificates shall become definitive.
- 459) If during this period, interoperability faults are found, the laboratory in charge of interoperability tests shall identify the causes of the problems with the help of all manufacturers involved and shall invite them to realize the necessary modifications.
- 460) If at the end of this period, interoperability problems still remain, the laboratory in charge of interoperability tests, with the collaboration of the manufacturers concerned and with the type approval authorities who delivered the corresponding functional certificates shall find out the causes of the interoperability faults and establish which modifications should be made by each of the manufacturers concerned. The search for technical solutions shall last for a maximum of two months, after which, if no common solution is found, the Commission, after having consulted the laboratory in charge of interoperability tests, shall decide which equipment(s) and cards get a definitive interoperability certificate and state the reasons why.
- 461) Any request for interoperability tests, registered by the laboratory between the end of the four month period after the first provisional interoperability certificate has been delivered and the date of the decision by the Commission referred to in requirement 455, shall be postponed until the initial interoperability problems have been solved. Those requests are then processed in the chronological order of their registration.

## ANNEX II

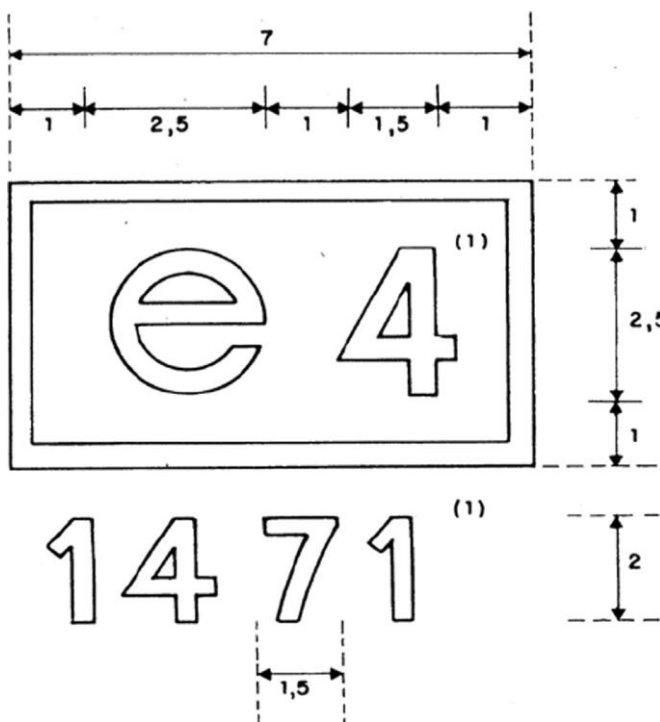
## APPROVAL MARK AND CERTIFICATE

## I. APPROVAL MARK

1. The approval mark shall be made up of:

Belgium	6,
Bulgaria	34,
Czech Republic	8,
Denmark	18,
Germany	1,
Estonia	29,
Ireland	24,
Greece	23,
Spain	9,
France	2,
Croatia	25,
Italy	3,
Cyprus	CY,
Latvia	32,
Lithuania	36,
Luxembourg	13,
Hungary	7,
Malta	MT,
Netherlands	4,
Austria	12,
Poland	20,

- (a) a Portugal 21,
- shall be Romania 19,
- or letter Slovenia 26,
- has Slovakia 27,
- Finland 17,
- signs: Sweden 5,
- United Kingdom 11,



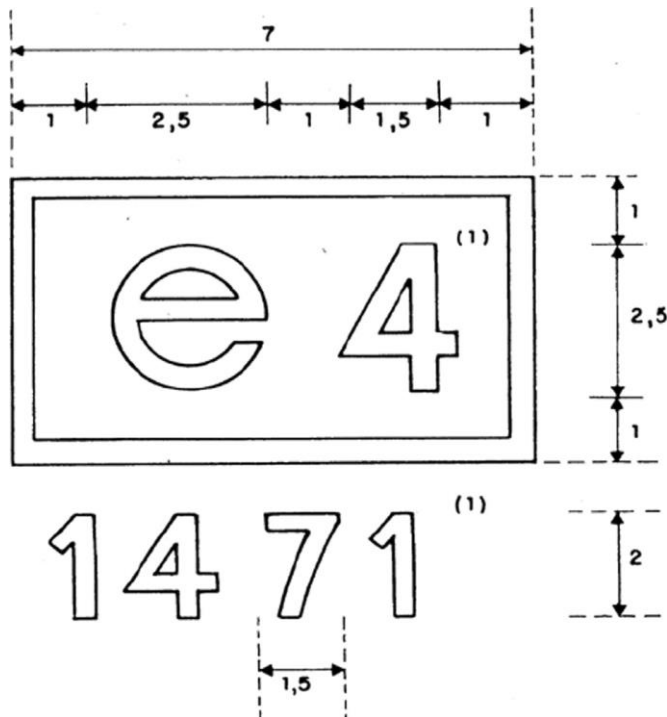
rectangle, within which placed the letter 'e' followed by a distinguishing number for the country which issued the approval in accordance with the following conventional

and

(b) an approval number corresponding to the number of the approval certificate drawn up for the prototype of the recording equipment or the record sheet or to the number of a tachograph card, placed at any point within the immediate proximity of that rectangle.

2. The approval mark shall be shown on the descriptive plaque of each set of equipment and on each record sheet and on each tachograph card. It must be indelible and must always remain clearly legible.

3. The dimensions of the approval mark drawn below <sup>(1)</sup> are expressed in millimetres, these dimensions being minima. The ratios between the dimensions must be maintained.



---

<sup>(1)</sup> These figures are shown for guidance only.

II. APPROVAL CERTIFICATE FOR ANALOGUE TACHOGRAPHS

A Member State which has granted approval shall issue the applicant with an approval certificate, the model of which is given below. When informing other Member States of approvals issued or, if the occasion should arise, withdrawn, a Member State shall use copies of that certificate.

APPROVAL CERTIFICATE

Name of competent administration .....

Notification concerning <sup>(1)</sup>:

- approval of a type of recording equipment
- withdrawal of approval of a type of recording equipment
- approval of a model record sheet
- withdrawal of approval of a model record sheet

Approval No: .....

1. Trade mark or name .....
2. Name of type or model .....
3. Name of manufacturer .....
4. Address of manufacturer .....
5. Submitted for approval on .....
6. Tested at .....
7. Date and number of the test(s) .....
8. Date of approval .....
9. Date of withdrawal of approval .....
10. Type or types of recording equipment in which sheet is designed to be used .....

11. Place .....

12. Date .....

13. Descriptive documents annexed .....

14. Remarks (including the position of seals if applicable)

(Signature)

(1) Delete items not applicable.



III. APPROVAL CERTIFICATE FOR DIGITAL TACHOGRAPHS

A Member State which has granted approval shall issue the applicant with an approval certificate, the model of which is given below. When informing other Member States of approvals issued or, if the occasion should arise, withdrawn, a Member State shall use copies of that certificate.

APPROVAL CERTIFICATE FOR DIGITAL TACHOGRAPHS

Name of competent administration .....

Notification concerning <sup>(1)</sup>:

- |                                       |   |
|---------------------------------------|---|
| <input type="checkbox"/> approval of: | <input type="checkbox"/> withdrawal of approval of:                   |
|                                       | <input type="checkbox"/> recording equipment model                    |
|                                       | <input type="checkbox"/> recording equipment component <sup>(2)</sup> |
|                                       | <input type="checkbox"/> a driver's card                              |
|                                       | <input type="checkbox"/> a workshop card                              |
|                                       | <input type="checkbox"/> a company card                               |
|                                       | <input type="checkbox"/> a controller's card                          |

Approval No: .....

1. Manufacturing brand or trademark .....
2. Name of model .....
3. Name of manufacturer .....
4. Address of manufacturer .....
5. Submitted for approval for .....
6. Laboratory(-ies) .....
7. Date and number of test report .....
8. Date of approval .....
9. Date of withdrawal of approval .....
10. Model of recording equipment(s) with which the component is designed to be used .....
11. Place .....

12. Date .....

13. Descriptive documents annexed .....

14. Remarks (including the position of seals if applicable)

(Signature)

(1) Tick the relevant boxes.

(2) Specify the component dealt with in the notification.

IV. APPROVAL CERTIFICATE FOR SMART TACHOGRAPHS

A Member State which has granted approval shall issue the applicant with an approval certificate, the model of which is given below. When informing other Member States of approvals issued or, if the occasion should arise, withdrawn, a Member State shall use copies of that certificate.

APPROVAL CERTIFICATE FOR SMART TACHOGRAPHS

Name of competent administration .....

Notification concerning <sup>(1)</sup>:

- |                                       |   |
|---------------------------------------|---|
| <input type="checkbox"/> approval of: | <input type="checkbox"/> withdrawal of approval of:                   |
|                                       | <input type="checkbox"/> recording equipment model                    |
|                                       | <input type="checkbox"/> recording equipment component <sup>(2)</sup> |
|                                       | <input type="checkbox"/> a driver's card                              |
|                                       | <input type="checkbox"/> a workshop card                              |
|                                       | <input type="checkbox"/> a company card                               |
|                                       | <input type="checkbox"/> a controller's card                          |

Approval No: .....

1. Manufacturing brand or trademark .....

2. Name of model .....

3. Name of manufacturer .....

4. Address of manufacturer .....

5. Submitted for approval for .....

- 6. a Test laboratory for functional certification .....
- b Test laboratory for security certification .....
- c Test laboratory for interoperability certification .....
- 7. a Date and number of functional certificate.....
- b Date and number of security certificate.....
- c Date and number of interoperability certificate .....
- 8. Date of approval .....
- 9. Date of withdrawal of approval .....
- 10. Model of recording equipment(s) with which the component is designed to be used .....
- 11. Place .....
- 12. Date .....
- 13. Descriptive documents annexed .....
- 14. Remarks (including the position of seals if applicable)

(Signature)

- (1) Tick the relevant boxes.
- (2) Specify the component dealt with in the notification.

## Appendix 1. Data dictionary

### TABLE OF CONTENT

<b>1. INTRODUCTION</b>	<b>109</b>
1.1. Approach for definitions of data types	109
1.2. References	109
<b>2. DATA TYPE DEFINITIONS</b>	<b>111</b>
2.1. ActivityChangeInfo	111
2.2. Address	112
2.3. AESKey	113
2.4. AES128Key	113
2.5. AES192Key	113
2.6. AES256Key	113
2.7. BCDString	114
2.8. CalibrationPurpose	114
2.9. CardActivityDailyRecord	115
2.10. CardActivityLengthRange	115
2.11. CardApprovalNumber	115
2.12. CardCertificate	116
2.13. CardChipIdentification	116
2.14. CardConsecutiveIndex	116
2.15. CardControlActivityDataRecord	117
2.16. CardCurrentUse	117
2.17. CardDriverActivity	118
2.18. CardDrivingLicenceInformation	118
2.19. CardEventData	119
2.20. CardEventRecord	119
2.21. CardFaultData	119
2.22. CardFaultRecord	120
2.23. CardIccIdentification	120
2.24. CardIdentification	121

---

2.25. CardMACertificate	121
2.26. CardNumber	122
2.27. CardPlaceDailyWorkPeriod	122
2.28. CardPrivateKey	123
2.29. CardPublicKey	123
2.30. CardRenewalIndex	123
2.31. CardReplacementIndex	123
2.32. CardSignCertificate	123
2.33. CardSlotNumber	123
2.34. CardSlotsStatus	124
2.35. CardSlotsStatusRecordArray	124
2.36. CardStructureVersion	125
2.37. CardVehicleRecord	125
2.38. CardVehiclesUsed	126
2.39. CardVehicleUnitRecord	127
2.40. CardVehicleUnitsUsed	127
2.41. Certificate	128
2.42. CertificateContent	128
2.43. CertificateHolderAuthorisation	129
2.44. CertificateRequestID	129
2.45. CertificationAuthorityKID	130
2.46. CompanyActivityData	131
2.47. CompanyActivityType	132
2.48. CompanyCardApplicationIdentification	132
2.49. CompanyCardHolderIdentification	132
2.50. ControlCardApplicationIdentification	133
2.51. ControlCardControlActivityData	133
2.52. ControlCardHolderIdentification	134
2.53. ControlType	135
2.54. CurrentDateTime	136

<b>2.55. CurrentDateTimeRecordArray</b>	<b>136</b>
<b>2.56. DailyPresenceCounter</b>	<b>137</b>
<b>2.57. Datef</b>	<b>137</b>
<b>2.58. DateOfDayDownloaded</b>	<b>138</b>
<b>2.59. DateOfDayDownloadedRecordArray</b>	<b>138</b>
<b>2.60. Distance</b>	<b>138</b>
<b>2.61. DriverCardApplicationIdentification</b>	<b>139</b>
<b>2.62. DriverCardHolderIdentification</b>	<b>140</b>
<b>2.63. DSRCSecurityData</b>	<b>140</b>
<b>2.64. EGFCertificate</b>	<b>141</b>
<b>2.65. EmbedderIcAssemblerId</b>	<b>141</b>
<b>2.66. EntryTypeDailyWorkPeriod</b>	<b>142</b>
<b>2.67. EquipmentType</b>	<b>143</b>
<b>2.68. EuropeanPublicKey</b>	<b>144</b>
<b>2.69. EventFaultRecordPurpose</b>	<b>145</b>
<b>2.70. EventFaultType</b>	<b>145</b>
<b>2.71. ExtendedSealIdentifier</b>	<b>147</b>
<b>2.72. ExtendedSerialNumber</b>	<b>148</b>
<b>2.73. FullCardNumber</b>	<b>149</b>
<b>2.74. FullCardNumberAndGeneration</b>	<b>149</b>
<b>2.75. Generation</b>	<b>149</b>
<b>2.76. GeoCoordinates</b>	<b>149</b>
<b>2.77. GNSSAccuracy</b>	<b>150</b>
<b>2.78. GNSSContinuousDriving</b>	<b>150</b>
<b>2.79. GNSSContinuousDrivingRecord</b>	<b>150</b>
<b>2.80. GNSSPlaceRecord</b>	<b>151</b>
<b>2.81. HighResOdometer</b>	<b>151</b>
<b>2.82. HighResTripDistance</b>	<b>151</b>
<b>2.83. HolderName</b>	<b>152</b>
<b>2.84. InternalGNSSReceiver</b>	<b>152</b>

---

<b>2.85. K-ConstantOfRecordingEquipment</b>	<b>152</b>
<b>2.86. KeyIdentifier</b>	<b>152</b>
<b>2.87. KMWCKey</b>	<b>152</b>
<b>2.88. Language</b>	<b>153</b>
<b>2.89. LastCardDownload</b>	<b>153</b>
<b>2.90. LinkCertificate</b>	<b>153</b>
<b>2.91. L-TyreCircumference</b>	<b>153</b>
<b>2.92. MAC</b>	<b>153</b>
<b>2.93. ManualInputFlag</b>	<b>154</b>
<b>2.94. ManufacturerCode</b>	<b>154</b>
<b>2.95. ManufacturerSpecificEventFaultData</b>	<b>154</b>
<b>2.96. MemberStateCertificate</b>	<b>154</b>
<b>2.97. MemberStateCertificateRecordArray</b>	<b>154</b>
<b>2.98. MemberStatePublicKey</b>	<b>155</b>
<b>2.99. Name</b>	<b>155</b>
<b>2.100. NationAlpha</b>	<b>156</b>
<b>2.101. NationNumeric</b>	<b>156</b>
<b>2.102. NoOfCalibrationRecords</b>	<b>156</b>
<b>2.103. NoOfCalibrationsSinceDownload</b>	<b>156</b>
<b>2.104. NoOfCardPlaceRecords</b>	<b>156</b>
<b>2.105. NoOfCardVehicleRecords</b>	<b>157</b>
<b>2.106. NoOfCardVehicleUnitRecords</b>	<b>157</b>
<b>2.107. NoOfCompanyActivityRecords</b>	<b>157</b>
<b>2.108. NoOfControlActivityRecords</b>	<b>157</b>
<b>2.109. NoOfEventsPerType</b>	<b>157</b>
<b>2.110. NoOfFaultsPerType</b>	<b>157</b>
<b>2.111. NoOfGNSSCDRecords</b>	<b>157</b>
<b>2.112. NoOfSpecificConditionRecords</b>	<b>158</b>
<b>2.113. OdometerShort</b>	<b>158</b>
<b>2.114. OdometerValueMidnight</b>	<b>158</b>

<b>2.115. OdometerValueMidnightRecordArray</b>	<b>158</b>
<b>2.116. OverspeedNumber</b>	<b>159</b>
<b>2.117. PlaceRecord</b>	<b>159</b>
<b>2.118. PreviousVehicleInfo</b>	<b>160</b>
<b>2.119. PublicKey</b>	<b>161</b>
<b>2.120. RecordType</b>	<b>161</b>
<b>2.121. RegionAlpha</b>	<b>162</b>
<b>2.122. RegionNumeric</b>	<b>164</b>
<b>2.123. RemoteCommunicationModuleSerialNumber</b>	<b>165</b>
<b>2.124. RSAKeyModulus</b>	<b>165</b>
<b>2.125. RSAKeyPrivateExponent</b>	<b>165</b>
<b>2.126. RSAKeyPublicExponent</b>	<b>165</b>
<b>2.127. RtmData</b>	<b>165</b>
<b>2.128. SealDataCard</b>	<b>165</b>
<b>2.129. SealDataVu</b>	<b>166</b>
<b>2.130. SealRecord</b>	<b>166</b>
<b>2.131. SensorApprovalNumber</b>	<b>166</b>
<b>2.132. SensorExternalGNSSApprovalNumber</b>	<b>166</b>
<b>2.133. SensorExternalGNSSCoupledRecord</b>	<b>167</b>
<b>2.134. SensorExternalGNSSIdentification</b>	<b>168</b>
<b>2.135. SensorExternalGNSSInstallation</b>	<b>168</b>
<b>2.136. SensorExternalGNSSOSIdentifier</b>	<b>169</b>
<b>2.137. SensorExternalGNSSSCIdentifier</b>	<b>169</b>
<b>2.138. SensorGNSSCouplingDate</b>	<b>169</b>
<b>2.139. SensorGNSSSerialNumber</b>	<b>170</b>
<b>2.140. SensorIdentification</b>	<b>170</b>
<b>2.141. SensorInstallation</b>	<b>171</b>
<b>2.142. SensorInstallationSecData</b>	<b>171</b>
<b>2.143. SensorOSIdentifier</b>	<b>172</b>
<b>2.144. SensorPaired</b>	<b>172</b>



---

<b>2.145. SensorPairedRecord</b>	<b>172</b>
<b>2.146. SensorPairingDate</b>	<b>173</b>
<b>2.147. SensorSCIdentifier</b>	<b>173</b>
<b>2.148. SensorSerialNumber</b>	<b>173</b>
<b>2.149. Signature</b>	<b>173</b>
<b>2.150. SignatureRecordArray</b>	<b>173</b>
<b>2.151. SimilarEventsNumber</b>	<b>174</b>
<b>2.152. SpecificConditionRecord</b>	<b>174</b>
<b>2.153. SpecificConditions</b>	<b>174</b>
<b>2.154. SpecificConditionType</b>	<b>175</b>
<b>2.155. Speed</b>	<b>175</b>
<b>2.156. SpeedAuthorised</b>	<b>176</b>
<b>2.157. SpeedAverage</b>	<b>176</b>
<b>2.158. SpeedMax</b>	<b>176</b>
<b>2.159. TachographPayload</b>	<b>176</b>
<b>2.160. TachographPayloadEncrypted</b>	<b>176</b>
<b>2.161. TDesSessionKey</b>	<b>177</b>
<b>2.162. TimeReal</b>	<b>177</b>
<b>2.163. TyreSize</b>	<b>177</b>
<b>2.164. VehicleIdentificationNumber</b>	<b>177</b>
<b>2.165. VehicleIdentificationNumberRecordArray</b>	<b>178</b>
<b>2.166. VehicleRegistrationIdentification</b>	<b>178</b>
<b>2.167. VehicleRegistrationNumber</b>	<b>178</b>
<b>2.168. VehicleRegistrationNumberRecordArray</b>	<b>178</b>
<b>2.169. VuAbility</b>	<b>179</b>
<b>2.170. VuActivityDailyData</b>	<b>179</b>
<b>2.171. VuActivityDailyRecordArray</b>	<b>180</b>
<b>2.172. VuApprovalNumber</b>	<b>180</b>
<b>2.173. VuCalibrationData</b>	<b>181</b>
<b>2.174. VuCalibrationRecord</b>	<b>181</b>

<b>2.175. VuCalibrationRecordArray</b>	<b>183</b>
<b>2.176. VuCardIWData</b>	<b>184</b>
<b>2.177. VuCardIWRecord</b>	<b>185</b>
<b>2.178. VuCardIWRecordArray</b>	<b>186</b>
<b>2.179. VuCardRecord</b>	<b>186</b>
<b>2.180. VuCardRecordArray</b>	<b>187</b>
<b>2.181. VuCertificate</b>	<b>187</b>
<b>2.182. VuCertificateRecordArray</b>	<b>187</b>
<b>2.183. VuCompanyLocksData</b>	<b>188</b>
<b>2.184. VuCompanyLocksRecord</b>	<b>189</b>
<b>2.185. VuCompanyLocksRecordArray</b>	<b>189</b>
<b>2.186. VuControlActivityData</b>	<b>190</b>
<b>2.187. VuControlActivityRecord</b>	<b>190</b>
<b>2.188. VuControlActivityRecordArray</b>	<b>191</b>
<b>2.189. VuDataBlockCounter</b>	<b>191</b>
<b>2.190. VuDetailedSpeedBlock</b>	<b>192</b>
<b>2.191. VuDetailedSpeedBlockRecordArray</b>	<b>192</b>
<b>2.192. VuDetailedSpeedData</b>	<b>192</b>
<b>2.193. VuDownloadablePeriod</b>	<b>193</b>
<b>2.194. VuDownloadablePeriodRecordArray</b>	<b>193</b>
<b>2.195. VuDownloadActivityData</b>	<b>194</b>
<b>2.196. VuDownloadActivityDataRecordArray</b>	<b>194</b>
<b>2.197. VuEventData</b>	<b>195</b>
<b>2.198. VuEventRecord</b>	<b>195</b>
<b>2.199. VuEventRecordArray</b>	<b>196</b>
<b>2.200. VuFaultData</b>	<b>197</b>
<b>2.201. VuFaultRecord</b>	<b>198</b>
<b>2.202. VuFaultRecordArray</b>	<b>199</b>
<b>2.203. VuGNSSCDRecord</b>	<b>199</b>
<b>2.204. VuGNSSCDRecordArray</b>	<b>200</b>

---

<b>2.205. VuIdentification</b>	<b>201</b>
<b>2.206. VuIdentificationRecordArray</b>	<b>202</b>
<b>2.207. VuITSConsentRecord</b>	<b>202</b>
<b>2.208. VuITSConsentRecordArray</b>	<b>202</b>
<b>2.209. VuManufacturerAddress</b>	<b>203</b>
<b>2.210. VuManufacturerName</b>	<b>203</b>
<b>2.211. VuManufacturingDate</b>	<b>203</b>
<b>2.212. VuOverSpeedingControlData</b>	<b>203</b>
<b>2.213. VuOverSpeedingControlDataRecordArray</b>	<b>204</b>
<b>2.214. VuOverSpeedingEventData</b>	<b>204</b>
<b>2.215. VuOverSpeedingEventRecord</b>	<b>205</b>
<b>2.216. VuOverSpeedingEventRecordArray</b>	<b>206</b>
<b>2.217. VuPartNumber</b>	<b>206</b>
<b>2.218. VuPlaceDailyWorkPeriodData</b>	<b>207</b>
<b>2.219. VuPlaceDailyWorkPeriodRecord</b>	<b>207</b>
<b>2.220. VuPlaceDailyWorkPeriodRecordArray</b>	<b>207</b>
<b>2.221. VuPrivateKey</b>	<b>208</b>
<b>2.222. VuPublicKey</b>	<b>208</b>
<b>2.223. VuSerialNumber</b>	<b>208</b>
<b>2.224. VuSoftInstallationDate</b>	<b>208</b>
<b>2.225. VuSoftwareIdentification</b>	<b>209</b>
<b>2.226. VuSoftwareVersion</b>	<b>209</b>
<b>2.227. VuSpecificConditionData</b>	<b>209</b>
<b>2.228. VuSpecificConditionRecordArray</b>	<b>209</b>
<b>2.229. VuTimeAdjustmentData</b>	<b>210</b>
<b>2.230. VuTimeAdjustmentGNSSRecord</b>	<b>210</b>
<b>2.231. VuTimeAdjustmentGNSSRecordArray</b>	<b>210</b>
<b>2.232. VuTimeAdjustmentRecord</b>	<b>211</b>
<b>2.233. VuTimeAdjustmentRecordArray</b>	<b>212</b>
<b>2.234. WorkshopCardApplicationIdentification</b>	<b>213</b>

<b>2.235. WorkshopCardCalibrationData</b>	<b>214</b>
<b>2.236. WorkshopCardCalibrationRecord</b>	<b>215</b>
<b>2.237. WorkshopCardHolderIdentification</b>	<b>217</b>
<b>2.238. WorkshopCardPIN</b>	<b>217</b>
<b>2.239. W-VehicleCharacteristicConstant</b>	<b>217</b>
<b>2.240. VuPowerSupplyInterruptionRecord</b>	<b>218</b>
<b>2.241. VuPowerSupplyInterruptionRecordArray</b>	<b>218</b>
<b>2.242. VuSensorExternalGNSSCoupledRecordArray</b>	<b>219</b>
<b>2.243. VuSensorPairedRecordArray</b>	<b>219</b>
<b>3. VALUE AND SIZE RANGE DEFINITIONS</b>	<b>220</b>
<b>4. CHARACTER SETS</b>	<b>220</b>
<b>5. ENCODING</b>	<b>221</b>
<b>6. OBJECT IDENTIFIERS UND APPLICATION IDENTIFIERS</b>	<b>221</b>
<b>6.1. Object Identifiers</b>	<b>221</b>
<b>6.2. Application Identifiers</b>	<b>223</b>

## 1. Introduction

This appendix specifies data formats, data elements, and data structures for use within the recording equipment and tachograph cards.

### 1.1. Approach for definitions of data types

This appendix uses Abstract Syntax Notation One (ASN.1) to define data types. This enables simple and structured data to be defined without implying any specific transfer syntax (encoding rules) which will be application and environment dependent.

ASN.1 type naming conventions are done in accordance with ISO/IEC 8824-1. This implies that:

- where possible, the meaning of the data type is implied through the names being selected,
- where a data type is a composition of other data types, the data type name is still a single sequence of alphabetical characters commencing with a capital letter, however capitals are used within the name to impart the corresponding meaning,
- in general, the data types names are related to the name of the data types from which they are constructed, the equipment in which data is stored and the function related to the data.

If an ASN.1 type is already defined as part of another standard and if it is relevant for usage in the recording equipment, then this ASN.1 type will be defined in this appendix.

To enable several types of encoding rules, some ASN.1 types in this appendix are constrained by value range identifiers. The value range identifiers are defined in paragraph 3 and Appendix 2.

### 1.2. References

The following references are used in this Appendix:

ISO 639	Code for the representation of names of languages. First Edition: 1988.
ISO 3166	Codes for the representation of names of countries and their subdivisions – Part 1: Country codes, 2013
ISO 3779	Road vehicles - Vehicle identification number (VIN) - Content and structure. 2009
ISO/IEC 7816-5	Identification cards - Integrated circuit cards - Part 5: Registration of application providers. Second edition: 2004.
ISO/IEC 7816-6	Identification cards - Integrated circuit cards – Part 6: Interindustry data elements for interchange, 2004 + Technical Corrigendum 1: 2006
ISO/IEC 8824-1	Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation. 2008 + Technical Corrigendum 1: 2012 and Technical Corrigendum 2: 2014.
ISO/IEC 8825-2	Information technology - ASN.1 encoding rules: Specification of Packed Encoding Rules (PER). 2008.
ISO/IEC 8859-1	Information technology - 8 bit single-byte coded graphic character sets - Part 1: Latin alphabet No.1. First edition: 1998.
ISO/IEC 8859-7	Information technology - 8 bit single-byte coded graphic character sets - Part 7: Latin/Greek alphabet. 2003.
ISO 16844-3	Road vehicles - Tachograph systems - Motion Sensor Interface. 2004 + Technical Corrigendum 1: 2006..

TR-03110-3      BSI / ANSSI Technical Guideline TR-03110-3, Advanced Security Mechanisms for Machine Readable Travel Documents and eIDAS Token - Part 3 Common Specifications, version 2.20, 3. February 2015

## 2. Data Type Definitions

For any of the following data types, the default value for an “unknown” or a “not applicable” content will consist in filling the data element with ‘FF’ bytes.

All data types are used for Generation 1 and Generation 2 applications unless otherwise specified.

### 2.1. ActivityChangeInfo

This data type enables to code, within a two bytes word, a slot status at 00:00 and/or a driver status at 00:00 and/or changes of activity and/or changes of driving status and/or changes of card status for a driver or a co-driver. This data type is related to Annex 1C requirements 105, 266, 291, 320, 321, 343, and 344.

ActivityChangeInfo ::= OCTET STRING (SIZE(2))

**Value assignment – Octet Aligned** : ‘scpaatttttttt’B (16 bits)

For Data Memory recordings (or slot status):

‘s’B	Slot:
	‘0’B: DRIVER,
	‘1’B: CO-DRIVER,
‘c’B	Driving status:
	‘0’B: SINGLE,
	‘1’B: CREW,
‘p’B	Driver (or workshop) card status in the relevant slot:
	‘0’B: INSERTED, a card is inserted,
	‘1’B: NOT INSERTED, no card is inserted (or a card is withdrawn),
‘aa’B	Activity:
	‘00’B: BREAK/REST,
	‘01’B: AVAILABILITY,
	‘10’B: WORK,
	‘11’B: DRIVING,
‘tttttttt’B	Time of the change: Number of minutes since 00h00 on the given day.

For Driver (or Workshop) card recordings (and driver status):

- 's'B Slot (not relevant when 'p'=1 except note below):  
'0'B: DRIVER,  
'1'B: CO-DRIVER,
- 'c'B Driving status (case 'p'=0) or Following activity status (case 'p'=1):  
'0'B: SINGLE, '0'B: UNKNOWN  
'1'B: CREW, '1'B: KNOWN (=manually entered)
- 'p'B Card status:  
'0'B: INSERTED, the card is inserted in a recording equipment,  
'1'B: NOT INSERTED, the card is not inserted (or the card is withdrawn),
- 'aa'B Activity (not relevant when 'p'=1 and 'c'=0 except note below):  
'00'B: BREAK/REST,  
'01'B: AVAILABILITY,  
'10'B: WORK,  
'11'B: DRIVING,
- 'tttttttt'B Time of the change: Number of minutes since 00h00 on the given day.

**Note for the case 'card withdrawal':**

When the card is withdrawn:

- 's' is relevant and indicates the slot from which the card is withdrawn,
- 'c' must be set to 0,
- 'p' must be set to 1,
- 'aa' must code the current activity selected at that time,

As a result of a manual entry, the bits 'c' and 'aa' of the word (stored in a card) may be overwritten later to reflect the entry.

## 2.2. Address

An address.

Address ::= SEQUENCE {

```
codePage          INTEGER (0..255),
address           OCTET STRING (SIZE(35))
}
```

**codePage** specifies a character set defined in Chapter 4,

**address** is an address encoded using the specified character set.



## 2.3. AESKey

**Generation 2:**

An AES key with a length of 128, 192 or 256 bits.

```
AESKey ::= CHOICE {
    aes128Key          AES128Key,
    aes192Key          AES192Key,
    aes256Key          AES256Key
}
```

**Value assignment:** not further specified.

## 2.4. AES128Key

**Generation 2:**

An AES128 key.

```
AES128Key ::= SEQUENCE {
    length              INTEGER(0..255),
    aes128Key           OCTET STRING (SIZE(16))
}
```

**length** denotes the length of the AES128 key in octets.

**aes128Key** is an AES key with a length of 128 bits.

**Value assignment:**

The length shall have the value 16.

## 2.5. AES192Key

**Generation 2:**

An AES192 key.

```
AES192Key ::= SEQUENCE {
    length              INTEGER(0..255),
    aes192Key           OCTET STRING (SIZE(24))
}
```

**length** denotes the length of the AES192 key in octets.

**aes192Key** is an AES key with a length of 192 bits.

**Value assignment:**

The length shall have the value 24.

## 2.6. AES256Key

**Generation 2:**

An AES256 key.

```
AES256Key ::= SEQUENCE {  
    length                INTEGER(0..255),  
    aes256Key             OCTET STRING (SIZE(32))  
}
```

**length** denotes the length of the AES256 key in octets.

**aes256Key** is an AES key with a length of 256 bits.

**Value assignment:**

The length shall have the value 32.

## 2.7. BCDString

BCDString is applied for Binary Code Decimal (BCD) representation. This data type is used to represent one decimal digit in one semi octet (4 bits). BCDString is based on the ISO/IEC 8824-1 'CharacterStringType'.

```
BCDString ::= CHARACTER STRING (WITH COMPONENTS {  
    identification ( WITH COMPONENTS {  
        fixed PRESENT }) })
```

BCDString uses an "hstring" notation. The leftmost hexadecimal digit shall be the most significant semi octet of the first octet. To produce a multiple of octets, zero trailing semi octets shall be inserted, as needed, from the leftmost semi octet position in the first octet.

Permitted digits are : 0, 1, .. 9.

## 2.8. CalibrationPurpose

Code explaining why a set of calibration parameters was recorded. This data type is related to Annex 1B requirements 097 and 098 and Annex 1C requirements 119.

```
CalibrationPurpose ::= OCTET STRING (SIZE(1))
```

**Value assignment:**

Generation 1:

- '00'H reserved value,
- '01'H activation: recording of calibration parameters known, at the moment of the VU activation,
- '02'H first installation: first calibration of the VU after its activation,
- '03'H installation: first calibration of the VU in the current vehicle,
- '04'H periodic inspection.

Generation 2:

In addition to generation 1 the following values are used:

- '05'H entry of VRN by company,

'06'H time adjustment without calibration,  
 '07'H to '7F'H RFU, '80'H to 'FF'H Manufacturer specific.

## 2.9. CardActivityDailyRecord

Information, stored in a card, related to the driver activities for a particular calendar day. This data type is related to Annex 1C requirements 266, 291, 320 and 343.

```
CardActivityDailyRecord ::= SEQUENCE {
    activityPreviousRecordLength    INTEGER(0..CardActivityLengthRange),
    activityRecordLength            INTEGER(0..CardActivityLengthRange),
    activityRecordDate              TimeReal,
    activityDailyPresenceCounter    DailyPresenceCounter,
    activityDayDistance             Distance,
    activityChangeInfo             SET SIZE(1..1440) OF ActivityChangeInfo
}
```

**activityPreviousRecordLength** is the total length in bytes of the previous daily record. The maximum value is given by the length of the OCTET STRING containing these records (see CardActivityLengthRange Appendix 2 paragraph 4). When this record is the oldest daily record, the value of activityPreviousRecordLength must be set to 0.

**activityRecordLength** is the total length in bytes of this record. The maximum value is given by the length of the OCTET STRING containing these records.

**activityRecordDate** is the date of the record.

**activityDailyPresenceCounter** is the daily presence counter for the card this day.

**activityDayDistance** is the total distance travelled this day.

**activityChangeInfo** is the set of ActivityChangeInfo data for the driver this day. It may contain at maximum 1440 values (one activity change per minute). This set always includes the activityChangeInfo coding the driver status at 00:00.

## 2.10. CardActivityLengthRange

Number of bytes in a driver or a workshop card, available to store driver activity records.

```
CardActivityLengthRange ::= INTEGER(0..216-1)
```

**Value assignment:** see Appendix 2.

## 2.11. CardApprovalNumber

Type approval number of the card.

```
CardApprovalNumber ::= IA5String(SIZE(8))
```

**Value assignment:**

The approval number shall be provided as published on the corresponding European Commission web site, i.e. for example including hyphens if any. The approval number shall be left-aligned.

## 2.12. CardCertificate

Generation 1:

Certificate of the public key of a card.

CardCertificate ::= Certificate

## 2.13. CardChipIdentification

Information, stored in a card, related to the identification of the card's Integrated Circuit (IC) (Annex 1C requirement 249). The icSerialNumber together with the icManufacturingReferences identifies the card chip uniquely. The icSerialNumber alone does not uniquely identify the card chip.

CardChipIdentification ::= SEQUENCE {

icSerialNumber OCTET STRING (SIZE(4)),

icManufacturingReferences OCTET STRING (SIZE(4))

}

**icSerialNumber** is the IC serial number.

**icManufacturingReferences** is the IC manufacturer specific identifier.

## 2.14. CardConsecutiveIndex

A card consecutive index (definition h)).

CardConsecutiveIndex ::= IA5String(SIZE(1))

**Value assignment:** (see Annex 1C chapter 7)

Order for increase: '0 , ..., 9, A , ... , Z , a , ... , z'

## 2.15. CardControlActivityDataRecord

Information, stored in a driver or workshop card, related to the last control the driver has been subject to (Annex 1C requirements 274, 299, 327, and 350).

```
CardControlActivityDataRecord ::= SEQUENCE {
    controlType          ControlType,
    controlTime          TimeReal,
    controlCardNumber    FullCardNumber,
    controlVehicleRegistration VehicleRegistrationIdentification,
    controlDownloadPeriodBegin TimeReal,
    controlDownloadPeriodEnd TimeReal
}
```

**controlType** is the type of the control.

**controlTime** is the date and time of the control.

**controlCardNumber** is the FullCardNumber of the control officer having performed the control.

**controlVehicleRegistration** is the VRN and registering Member State of the vehicle in which the control happened.

**controlDownloadPeriodBegin** and **controlDownloadPeriodEnd** is the period downloaded, in case of downloading.

## 2.16. CardCurrentUse

Information about the actual usage of the card (Annex 1C requirement 273, 298, 326, and 349).

```
CardCurrentUse ::= SEQUENCE {
    sessionOpenTime      TimeReal,
    sessionOpenVehicle    VehicleRegistrationIdentification
}
```

**sessionOpenTime** is the time when the card is inserted for the current usage. This element is set to zero at card removal.

**sessionOpenVehicle** is the identification of the currently used vehicle, set at card insertion. This element is set to zero at card removal.

### 2.17. CardDriverActivity

Information, stored in a driver or a workshop card, related to the activities of the driver (Annex 1C requirements 267, 268, 292, 293, 321 and 344).

```
CardDriverActivity ::= SEQUENCE {  
    activityPointerOldestDayRecord      INTEGER(0.. CardActivityLengthRange-1),  
    activityPointerNewestRecord         INTEGER(0.. CardActivityLengthRange-1),  
    activityDailyRecords                OCTET STRING (SIZE(CardActivityLengthRange))  
}
```

**activityPointerOldestDayRecord** is the specification of the begin of the storage place (number of bytes from the beginning of the string) of the oldest complete day record in the activityDailyRecords string. The maximum value is given by the length of the string.

**activityPointerNewestRecord** is the specification of the begin of the storage place (number of bytes from the beginning of the string) of the most recent day record in the activityDailyRecords string. The maximum value is given by the length of the string.

**activityDailyRecords** is the space available to store the driver activity data (data structure: CardActivityDailyRecord) for each calendar day where the card has been used.

**Value assignment:** this octet string is cyclically filled with records of CardActivityDailyRecord. At the first use storing is started at the first byte of the string. All new records are appended at the end of the previous one. When the string is full, storing continues at the first byte of the string independently of a break being inside a data element. Before placing new activity data in the string (enlarging current activityDailyRecord, or placing a new activityDailyRecord) that replaces older activity data, activityPointerOldestDayRecord must be updated to reflect the new location of the oldest complete day record, and activityPreviousRecordLength of this (new) oldest complete day record must be reset to 0.

### 2.18. CardDrivingLicenceInformation

Information, stored in a driver card, related to the card holder driver licence data (Annex 1C requirement 259 and 284).

```
CardDrivingLicenceInformation ::= SEQUENCE {  
    drivingLicenceIssuingAuthority      Name,  
    drivingLicenceIssuingNation         NationNumeric,  
    drivingLicenceNumber                IA5String(SIZE(16))  
}
```

**drivingLicenceIssuingAuthority** is the authority responsible for issuing the driving licence.

**drivingLicenceIssuingNation** is the nationality of the authority that issued the driving licence.

**drivingLicenceNumber** is the number of the driving licence.

### 2.19. CardEventData

Information, stored in a driver or workshop card, related to the events associated with the card holder (Annex 1C requirements 260, 285, 318 and 341).

```
CardEventData ::= SEQUENCE SIZE(6) OF {  
    cardEventRecords          SET SIZE(NoOfEventsPerType) OF CardEventRecord  
}
```

**CardEventData** is a sequence, ordered by ascending value of EventFaultType, of cardEventRecords (except security breach attempts related records which are gathered in the last set of the sequence).

**cardEventRecords** is a set of event records of a given event type (or category for security breach attempts events).

### 2.20. CardEventRecord

Information, stored in a driver or a workshop card, related to an event associated to the card holder (Annex 1C requirements 261, 286, 318 and 341).

```
CardEventRecord ::= SEQUENCE {  
    eventType                  EventFaultType,  
    eventBeginTime            TimeReal,  
    eventEndTime              TimeReal,  
    eventVehicleRegistration   VehicleRegistrationIdentification  
}
```

**eventType** is the type of the event.

**eventBeginTime** is the date and time of beginning of event.

**eventEndTime** is the date and time of end of event.

**eventVehicleRegistration** is the VRN and registering Member State of vehicle in which the event happened.

### 2.21. CardFaultData

Information, stored in a driver or a workshop card, related to the faults associated to the card holder (Annex 1C requirements 263, 288, 318, and 341).

```
CardFaultData ::= SEQUENCE SIZE(2) OF {  
    cardFaultRecords          SET SIZE(NoOfFaultsPerType) OF CardFaultRecord  
}
```

**CardFaultData** is a sequence of Recording Equipment faults set of records followed by card faults set of records.

**cardFaultRecords** is a set of fault records of a given fault category (Recording Equipment or card).

## 2.22. CardFaultRecord

Information, stored in a driver or a workshop card, related to a fault associated to the card holder (Annex 1C requirement 264, 289, 318, and 341).

```
CardFaultRecord ::= SEQUENCE {  
    faultType                EventFaultType,  
    faultBeginTime           TimeReal,  
    faultEndTime             TimeReal,  
    faultVehicleRegistration VehicleRegistrationIdentification  
}
```

**faultType** is the type of the fault.

**faultBeginTime** is the date and time of beginning of fault.

**faultEndTime** is the date and time of end of fault.

**faultVehicleRegistration** is the VRN and registering Member State of vehicle in which the fault happened.

## 2.23. CardIccIdentification

Information, stored in a card, related to the identification of the integrated circuit (IC) card (Annex 1C requirement 248).

```
CardIccIdentification ::= SEQUENCE {  
    clockStop                OCTET STRING (SIZE(1)),  
    cardExtendedSerialNumber ExtendedSerialNumber,  
    cardApprovalNumber       CardApprovalNumber,  
    cardPersonaliserID       ManufacturerCode,  
    embedderIcAssemblerId    EmbedderIcAssemblerId,  
    icIdentifier              OCTET STRING (SIZE(2))  
}
```

**clockStop** is the Clockstop mode as defined in appendix 2.

**cardExtendedSerialNumber** is the IC card unique serial number as further specified by the ExtendedSerialNumber data type.

**cardApprovalNumber** is the type approval number of the card.

**cardPersonaliserID** is the card personaliser ID encoded as ManufacturerCode.

**embedderIcAssemblerId** provides information about the embedder/IC assembler.

**icIdentifier** is the Identifier of the IC on the card and its IC manufacturer as defined in ISO/IEC 7816-6.



#### 2.24. CardIdentification

Information, stored in a card, related to the identification of the card (Annex 1C requirements 255, 280, 310, 333, 359, 365, 371, and 377).

```
CardIdentification ::= SEQUENCE {  
    cardIssuingMemberState      NationNumeric,  
    cardNumber                  CardNumber,  
    cardIssuingAuthorityName    Name,  
    cardIssueDate               TimeReal,  
    cardValidityBegin           TimeReal,  
    cardExpiryDate              TimeReal  
}
```

**cardIssuingMemberState** is the code of the Member State issuing the card.

**cardNumber** is the card number of the card.

**cardIssuingAuthorityName** is the name of the authority having issued the Card.

**cardIssueDate** is the issue date of the Card to the current holder.

**cardValidityBegin** is the first date of validity of the card.

**cardExpiryDate** is the date when the validity of the card ends.

#### 2.25. CardMACertificate

Generation 2:

Certificate of the card public key for mutual authentication with a VU. The structure of this certificate is specified in Appendix 11.

CardMACertificate ::= Certificate

## 2.26. CardNumber

A card number as defined by definition g).

```
CardNumber ::= CHOICE {  
  SEQUENCE {  
    driverIdentification          IA5String(SIZE(14)),  
    cardReplacementIndex        CardReplacementIndex,  
    cardRenewalIndex            CardRenewalIndex  
  },  
  SEQUENCE {  
    ownerIdentification          IA5String(SIZE(13)),  
    cardConsecutiveIndex        CardConsecutiveIndex,  
    cardReplacementIndex        CardReplacementIndex,  
    cardRenewalIndex            CardRenewalIndex  
  }  
}
```

**driverIdentification** is the unique identification of a driver in a Member State.

**ownerIdentification** is the unique identification of a company or a workshop or a control body within a member state.

**cardConsecutiveIndex** is the card consecutive index.

**cardReplacementIndex** is the card replacement index.

**cardRenewalIndex** is the card renewal index.

The first sequence of the choice is suitable to code a driver card number, the second sequence of the choice is suitable to code workshop, control, and company card numbers.

## 2.27. CardPlaceDailyWorkPeriod

Information, stored in a driver or a workshop card, related to the places where daily work periods begin and/or end (Annex 1C requirements 272, 297, 325, and 348).

```
CardPlaceDailyWorkPeriod ::= SEQUENCE {  
  placePointerNewestRecord      INTEGER(0 .. NoOfCardPlaceRecords-1),  
  placeRecords                  SET SIZE(NoOfCardPlaceRecords) OF PlaceRecord  
}
```

**placePointerNewestRecord** is the index of the last updated place record.

**Value assignment:** Number corresponding to the numerator of the place record, beginning with '0' for the first occurrence of the place records in the structure.

**placeRecords** is the set of records containing the information related to the places entered.

## 2.28. CardPrivateKey

Generation 1:

The private key of a card.

CardPrivateKey ::= RSAKeyPrivateExponent

## 2.29. CardPublicKey

The public key of a card.

CardPublicKey ::= PublicKey

## 2.30. CardRenewalIndex

A card renewal index (definition i)).

CardRenewalIndex ::= IA5String(SIZE(1))

**Value assignment:** (see this Annex chapter VII).

‘0’ First issue.

Order for increase: ‘0 , ... , 9 , A , ... , Z’

## 2.31. CardReplacementIndex

A card replacement index (definition j)).

CardReplacementIndex ::= IA5String(SIZE(1))

**Value assignment:** (see this Annex chapter VII).

‘0’ Original card.

Order for increase: ‘0 , ... , 9 , A , ... , Z’

## 2.32. CardSignCertificate

Generation 2:

Certificate of the card public key for signature. The structure of this certificate is specified in Appendix 11.

CardSignCertificate ::= Certificate

## 2.33. CardSlotNumber

Code to distinguish between the two slots of a Vehicle Unit.

```
CardSlotNumber ::= INTEGER {
    driverSlot                (0),
    co-driverSlot             (1)
}
```

**Value assignment:** not further specified.

#### 2.34. CardSlotsStatus

Code indicating the type of cards inserted in the two slots of the vehicle unit.

CardSlotsStatus ::= OCTET STRING (SIZE(1))

**Value assignment – Octet Aligned** : ‘ccccddd’B

‘cccc’B Identification of the type of card inserted in the co-driver slot,

‘ddd’B Identification of the type of card inserted in the driver slot,

with the following identification codes:

‘0000’B no card is inserted,

‘0001’B a driver card is inserted,

‘0010’B a workshop card is inserted,

‘0011’B a control card is inserted,

‘0100’B a company card is inserted.

#### 2.35. CardSlotsStatusRecordArray

Generation 2:

The CardSlotsStatus plus metadata as used in the download protocol.

CardSlotsStatusRecordArray ::= SEQUENCE {

recordType	RecordType,
recordSize	INTEGER(1..65535),
noOfRecords	INTEGER(0..65535),
records	SET SIZE(noOfRecords) OF CardSlotsStatus

}

**recordType** denotes the type of the record (CardSlotsStatus). **Value Assignment:** See RecordType

**recordSize** is the size of the CardSlotsStatus in bytes.

**noOfRecords** is the number of records in the set records.

**records** is the set of CardSlotsStatus records.

## 2.36. CardStructureVersion

Code indicating the version of the implemented structure in a tachograph card.

CardStructureVersion ::= OCTET STRING (SIZE(2))

**Value assignment:** 'aabb'H:

'aa'H Index for changes of the structure.

'00'H for Generation 1 applications

'01'H for Generation 2 applications

'bb'H Index for changes concerning the use of the data elements defined for the structure given by the high byte.

'00'H for this version of Generation 1 applications

'00'H for this version of Generation 2 applications

## 2.37. CardVehicleRecord

Information, stored in a driver or workshop card, related to a period of use of a vehicle during a calendar day (Annex 1C requirements 269, 294, 322, and 345).

Generation 1:

CardVehicleRecord ::= SEQUENCE {

vehicleOdometerBegin	OdometerShort,
vehicleOdometerEnd	OdometerShort,
vehicleFirstUse	TimeReal,
vehicleLastUse	TimeReal,
vehicleRegistration	VehicleRegistrationIdentification,
vuDataBlockCounter	VuDataBlockCounter

}

**vehicleOdometerBegin** is the vehicle odometer value at the beginning of the period of use of the vehicle.

**vehicleOdometerEnd** is the vehicle odometer value at the end of the period of use of the vehicle.

**vehicleFirstUse** is the date and time of the beginning of the period of use of the vehicle.

**vehicleLastUse** is the date and time of the end of the period of use of the vehicle.

**vehicleRegistration** is the VRN and the registering Member State of the vehicle.

**vuDataBlockCounter** is the value of the VuDataBlockCounter at last extraction of the period of use of the vehicle.

Generation 2:

```
CardVehicleRecord ::= SEQUENCE {  
    vehicleOdometerBegin           OdometerShort,  
    vehicleOdometerEnd            OdometerShort,  
    vehicleFirstUse                TimeReal,  
    vehicleLastUse                 TimeReal,  
    vehicleRegistration            VehicleRegistrationIdentification,  
    vuDataBlockCounter            VuDataBlockCounter,  
    vehicleIdentificationNumber    VehicleIdentificationNumber  
}
```

In addition to generation 1 the following data element is used:

**VehicleIdentificationNumber** is the vehicle identification number referring to the vehicle as a whole.

### 2.38. CardVehiclesUsed

Information, stored in a driver or workshop card, related to the vehicles used by the card holder (Annex 1C requirements 270, 295, 323, and 346).

```
CardVehiclesUsed ::= SEQUENCE {  
    vehiclePointerNewestRecord     INTEGER(0..NoOfCardVehicleRecords-1),  
    cardVehicleRecords             SET SIZE(NoOfCardVehicleRecords) OF CardVehicleRecord  
}
```

**vehiclePointerNewestRecord** is the index of the last updated vehicle record.

**Value assignment:** Number corresponding to the numerator of the vehicle record, beginning with '0' for the first occurrence of the vehicle records in the structure.

**cardVehicleRecords** is the set of records containing information on vehicles used.

## 2.39. CardVehicleUnitRecord

Generation 2:

Information, stored in a driver or workshop card, related to a vehicle unit that was used (Annex 1C requirement 303 and 351).

CardVehicleUnitRecord ::= SEQUENCE {

timeStamp	TimeReal,
manufacturerCode	ManufacturerCode,
deviceID	INTEGER(0..255),
vuSoftwareVersion	VuSoftwareVersion

}

**timeStamp** is the beginning of the period of use of the vehicle unit (i.e. first card insertion in the vehicle unit for the period).

**manufacturerCode** identifies the manufacturer of the Vehicle Unit.

**deviceID** identifies the Vehicle Unit type of a manufacturer. The value is manufacturer specific.

**vuSoftwareVersion** is the software version number of the Vehicle Unit.

## 2.40. CardVehicleUnitsUsed

Generation 2:

Information, stored in a driver or workshop card, related to the vehicle units used by the card holder (Annex 1C requirement 306 and 352).

CardVehicleUnitsUsed ::= SEQUENCE {

vehicleUnitPointerNewestRecord	INTEGER(0..NoOfCardVehicleUnitRecords-1),
cardVehicleUnitRecords	SET SIZE(NoOfCardVehicleUnitRecords) OF CardVehicleUnitRecord

}

**vehicleUnitPointerNewestRecord** is the index of the last updated vehicle unit record.

**Value assignment:** Number corresponding to the numerator of the vehicle unit record, beginning with '0' for the first occurrence of the vehicle unit records in the structure.

**cardVehicleUnitRecords** is the set of records containing information on vehicle units used.

## 2.41. Certificate

The certificate of a public key issued by a Certification Authority.

Generation 1:

Certificate ::= OCTET STRING (SIZE(194))

**Value assignment:** digital signature with partial recovery of a CertificateContent according to Appendix 11 common security mechanisms: Signature (128 bytes) || Public Key remainder (58 bytes) || Certification Authority Reference (8 bytes).

Generation 2:

Certificate ::= OCTET STRING (SIZE(204..341))

Value assignment: See Appendix 11

## 2.42. CertificateContent

Generation 1:

The (clear) content of the certificate of a public key according to Appendix 11 common security mechanisms.

CertificateContent ::= SEQUENCE {

certificateProfileIdentifier	INTEGER(0..255),
certificationAuthorityReference	KeyIdentifier,
certificateHolderAuthorisation	CertificateHolderAuthorisation,
certificateEndOfValidity	TimeReal,
certificateHolderReference	KeyIdentifier,
publicKey	PublicKey

}

**certificateProfileIdentifier** is the version of the corresponding certificate.

**Value assignment:** '01h' for this version.

**certificationAuthorityReference** identifies the Certification Authority issuing the certificate. It also references the Public Key of this Certification Authority.

**certificateHolderAuthorisation** identifies the rights of the certificate holder.

**certificateEndOfValidity** is the date when the certificate expires administratively.

**certificateHolderReference** identifies the certificate holder. It also references his Public Key.

**publicKey** is the public key that is certified by this certificate.



## 2.43. CertificateHolderAuthorisation

Identification of the rights of a certificate holder.

```
CertificateHolderAuthorisation ::= SEQUENCE {
    tachographApplicationID      OCTET STRING(SIZE(6))
    equipmentType                 EquipmentType
}
```

Generation 1:

**tachographApplicationID** is the application identifier for the tachograph application.

**Value assignment:** 'FFh' '54h' '41h' '43h' '48h' '4Fh'. This AID is a proprietary non registered application identifier in accordance with ISO/IEC 7816-5.

**equipmentType** is the identification of the type of equipment to which the certificate is intended.

**Value assignment:** in accordance with EquipmentType data type. **0** if certificate is the one of a Member State.

Generation 2:

**tachographApplicationID** denotes the 6 most significant bytes of the generation 2 tachograph card application identifier (AID). The AID for the tachograph card application is specified in chapter 6.2.

**Value assignment:** 'FF 53 4D 52 44 54'.

**equipmentType** is the identification of the type of equipment as specified for generation 2 to which the certificate is intended.

**Value assignment:** in accordance with EquipmentType data type.

## 2.44. CertificateRequestID

Unique identification of a certificate request. It can also be used as a Vehicle Unit Public Key Identifier if the serial number of the vehicle Unit to which the key is intended is not known at certificate generation time.

```
CertificateRequestID ::= SEQUENCE{
    requestSerialNumber          INTEGER(0..232-1),
    requestMonthYear             BCDSString(SIZE(2)),
    crIdentifier                 OCTET STRING(SIZE(1)),
    manufacturerCode            ManufacturerCode
}
```

**requestSerialNumber** is a serial number for the certificate request, unique for the manufacturer and the month below.

**requestMonthYear** is the identification of the month and the year of the certificate request.

**Value assignment:** BCD coding of Month (two digits) and Year (two last digits).

**crIdentifier:** is an identifier to distinguish a certificate request from an extended serial number.

**Value assignment:** 'FFh'.

**manufacturerCode:** is the numerical code of the manufacturer requesting the certificate.

#### 2.45. CertificationAuthorityKID

Identifier of the Public Key of a Certification Authority (a Member State or the European Certification Authority).

```
CertificationAuthorityKID ::= SEQUENCE{  
    nationNumeric          NationNumeric,  
    nationAlpha           NationAlpha,  
    keySerialNumber       INTEGER(0..255),  
    additionalInfo        OCTET STRING(SIZE(2)),  
    caIdentifier           OCTET STRING(SIZE(1))  
}
```

**nationNumeric** is the numerical nation code of the Certification Authority.

**nationAlpha** is the alphanumerical nation code of the Certification Authority.

**keySerialNumber** is a serial number to distinguish the different keys of the Certification Authority in the case keys are changed.

**additionalInfo** is a two byte field for additional coding (Certification Authority specific).

**caIdentifier** is an identifier to distinguish a Certification Authority Key Identifier from other Key Identifiers.

**Value assignment:** '01h'.

## 2.46. CompanyActivityData

Information, stored in a company card, related to activities performed with the card (Annex 1C requirement 373 and 379).

```

CompanyActivityData ::= SEQUENCE {
  companyPointerNewestRecord      INTEGER(0..NoOfCompanyActivityRecords-1),
  companyActivityRecords          SET SIZE(NoOfCompanyActivityRecords) OF
  companyActivityRecord           SEQUENCE {
    companyActivityType           CompanyActivityType,
    companyActivityTime           TimeReal,
    cardNumberInformation         FullCardNumber,
    vehicleRegistrationInformation VehicleRegistrationIdentification,
    downloadPeriodBegin          TimeReal,
    downloadPeriodEnd            TimeReal
  }
}

```

**companyPointerNewestRecord** is the index of the last updated companyActivityRecord.

**Value assignment:** Number corresponding to the numerator of the company activity record, beginning with '0' for the first occurrence of the company activity record in the structure.

**companyActivityRecords** is the set of all company activity records.

**companyActivityRecord** is the sequence of information related to one company activity.

**companyActivityType** is the type of the company activity.

**companyActivityTime** is the date and time of the company activity.

**cardNumberInformation** is the card number and the card issuing Member State of the card downloaded, if any.

**vehicleRegistrationInformation** is the VRN and registering Member State of the vehicle downloaded or locked in or out.

**downloadPeriodBegin** and **downloadPeriodEnd** is the period downloaded from the VU, if any.

## 2.47. CompanyActivityType

Code indicating an activity carried out by a company using its company card.

```
CompanyActivityType ::= INTEGER {  
    card downloading           (1),  
    VU downloading            (2),  
    VU lock-in                 (3),  
    VU lock-out                (4)  
}
```

## 2.48. CompanyCardApplicationIdentification

Information, stored in a company card related to the identification of the application of the card (Annex 1C requirement 369 and 375).

```
CompanyCardApplicationIdentification ::= SEQUENCE {  
    typeOfTachographCardId      EquipmentType,  
    cardStructureVersion         CardStructureVersion,  
    noOfCompanyActivityRecords   NoOfCompanyActivityRecords  
}
```

**typeOfTachographCardId** is specifying the implemented type of card.

**cardStructureVersion** is specifying the the version of the structure that is implemented in the card.

**noOfCompanyActivityRecords** is the number of company activity records the card can store.

## 2.49. CompanyCardHolderIdentification

Information, stored in a company card, related to the cardholder identification (Annex 1C requirement 372 and 378).

```
CompanyCardHolderIdentification ::= SEQUENCE {  
    companyName                  Name,  
    companyAddress                Address,  
    cardHolderPreferredLanguage   Language  
}
```

**companyName** is the name of the holder company.

**companyAddress** is the address of the holder company.

**cardHolderPreferredLanguage** is the preferred language of the card holder.

## 2.50. ControlCardApplicationIdentification

Information, stored in a control card related to the identification of the application of the card (Annex 1C requirement 357 and 363).

```
ControlCardApplicationIdentification ::= SEQUENCE {
    typeOfTachographCardId          EquipmentType,
    cardStructureVersion             CardStructureVersion,
    noOfControlActivityRecords       NoOfControlActivityRecords
}
```

**typeOfTachographCardId** is specifying the implemented type of card.

**cardStructureVersion** is specifying the version of the structure that is implemented in the card.

**noOfControlActivityRecords** is the number of control activity records the card can store.

## 2.51. ControlCardControlActivityData

Information, stored in a control card, related to control activity performed with the card (Annex 1C requirement 361 and 367).

```
ControlCardControlActivityData ::= SEQUENCE {
    controlPointerNewestRecord       INTEGER(0.. NoOfControlActivityRecords-1),
    controlActivityRecords           SET SIZE(NoOfControlActivityRecords) OF
    controlActivityRecord           SEQUENCE {
        controlType                  ControlType,
        controlTime                   TimeReal,
        controlledCardNumber          FullCardNumber,
        controlledVehicleRegistration VehicleRegistrationIdentification,
        controlDownloadPeriodBegin    TimeReal,
        controlDownloadPeriodEnd      TimeReal
    }
}
```

**controlPointerNewestRecord** is the index of the last updated control activity record.

**Value assignment:** Number corresponding to the numerator of the control activity record, beginning with '0' for the first occurrence of the control activity record in the structure.

**controlActivityRecords** is the set of all control activity records.

**controlActivityRecord** is the sequence of information related to one control.

**controlType** is the type of the control.

**controlTime** is the date and time of the control.

**controlledCardNumber** is the card number and the card issuing Member State of the card controlled.

**controlledVehicleRegistration** is the VRN and registering Member State of the vehicle in which the control happened.

**controlDownloadPeriodBegin** and **controlDownloadPeriodEnd** is the period eventually downloaded.

#### 2.52. ControlCardHolderIdentification

Information, stored in a control card, related to the identification of the cardholder (Annex 1C requirement 360 and 366).

```
ControlCardHolderIdentification ::= SEQUENCE {  
    controlBodyName           Name,  
    controlBodyAddress        Address,  
    cardHolderName            HolderName,  
    cardHolderPreferredLanguage Language  
}
```

**controlBodyName** is the name of the control body of the card holder.

**controlBodyAddress** is the address of the control body of the card holder.

**cardHolderName** is the name and first name(s) of the holder of the Control Card.

**cardHolderPreferredLanguage** is the preferred language of the card holder.

### 2.53. ControlType

Code indicating the activities carried out during a control. This data type is related to Annex 1C requirements 126, 274, 299, 327, and 350.

ControlType ::= OCTET STRING (SIZE(1))

Generation 1:

**Value assignment – Octet aligned** : ‘c’ B (8 bits)

‘c’ B card downloading:

‘0’ B: card not downloaded during this control activity,

‘1’ B: card downloaded during this control activity

‘v’ B VU downloading:

‘0’ B: VU not downloaded during this control activity,

‘1’ B: VU downloaded during this control activity

‘p’ B printing:

‘0’ B: no printing done during this control activity,

‘1’ B: printing done during this control activity

‘d’ B display:

‘0’ B: no display used during this control activity,

‘1’ B: display used during this control activity

‘xxxx’ B Not used.

Generation 2:

**Value assignment – Octet aligned** : ‘cvpdxxxx’B (8 bits)

‘c’B card downloading:

‘0’B: card not downloaded during this control activity,

‘1’B: card downloaded during this control activity

‘v’B VU downloading:

‘0’B: VU not downloaded during this control activity,

‘1’B: VU downloaded during this control activity

‘p’B printing:

‘0’B: no printing done during this control activity,

‘1’B: printing done during this control activity

‘d’B display:

‘0’B: no display used during this control activity,

‘1’B: display used during this control activity

‘e’B roadside calibration checking:

‘0’B: calibration parameters not checked during this control activity,

‘1’B: calibration parameters checked during this control activity

‘xxx’BRFU.

2.54. CurrentDateTime

The current date and time of the recording equipment.

CurrentDateTime ::= TimeReal

**Value assignment:** not further specified.

2.55. CurrentDateTimeRecordArray

Generation 2:

The current date and time plus metadata as used in the download protocol.



```

CurrentDateTimeRecordArray ::= SEQUENCE {
    recordType          RecordType,
    recordSize          INTEGER(1..65535),
    noOfRecords         INTEGER(0..65535),
    records              SET SIZE(noOfRecords) OF CurrentDateTime
}

```

**recordType** denotes the type of the record (CurrentDateTime). **Value Assignment:** See RecordType

**recordSize** is the size of the CurrentDateTime in bytes.

**noOfRecords** is the number of records in the set records.

**records** is a set of current date and time records.

#### 2.56. DailyPresenceCounter

Counter, stored in a driver or workshop card, increased by one for each calendar day the card has been inserted in a VU. This data type is related to Annex 1C requirements 266, 299, 320, and 343.

```
DailyPresenceCounter ::= BCDString(SIZE(2))
```

**Value assignment:** Consecutive Number with maximum value = 9 999, starting again with 0. At the time of first issuing of the card the number is set to 0.

#### 2.57. Datef

Date expressed in a readily printable numeric format.

```

Datef ::= SEQUENCE {
    year      BCDString(SIZE(2)),
    month     BCDString(SIZE(1)),
    day       BCDString(SIZE(1))
}

```

**Value assignment:**

yyyy Year

mm Month

dd Day

'00000000'H denotes explicitly no date.

## 2.58. DateOfDayDownloaded

Generation 2:

The date and time of the download.

DateOfDayDownloaded ::= TimeReal

**Value assignment:** not further specified.

## 2.59. DateOfDayDownloadedRecordArray

Generation 2:

The date and time of the download plus metadata as used in the download protocol.

DateOfDayDownloadedRecordArray ::= SEQUENCE {

recordType	RecordType,
recordSize	INTEGER(1..65535),
noOfRecords	INTEGER(0..65535),
records	SET SIZE(noOfRecords) OF DateOfDayDownloaded

}

**recordType** denotes the type of the record (DateOfDayDownloaded). **Value Assignment:** See RecordType

**recordSize** is the size of the CurrentDateTime in bytes.

**noOfRecords** is the number of records in the set records.

**records** is the set of date and time of the download records.

## 2.60. Distance

A distance travelled (result of the calculation of the difference between two vehicle's odometer values in kilometers).

Distance ::= INTEGER (0.. $2^{16}-1$ )

**Value assignment:** Unsigned binary. Value in km in the operational range 0 to 9 999 km.

## 2.61. DriverCardApplicationIdentification

Information, stored in a driver card related to the identification of the application of the card (Annex 1C requirement 253 and 278).

Generation 1:

```
DriverCardApplicationIdentification ::= SEQUENCE {
    typeOfTachographCardId      EquipmentType,
    cardStructureVersion         CardStructureVersion,
    noOfEventsPerType           NoOfEventsPerType,
    noOfFaultsPerType           NoOfFaultsPerType,
    activityStructureLength      CardActivityLengthRange,
    noOfCardVehicleRecords      NoOfCardVehicleRecords,
    noOfCardPlaceRecords        NoOfCardPlaceRecords
}
```

**typeOfTachographCardId** is specifying the implemented type of card.

**cardStructureVersion** is specifying the the version of the structure that is implemented in the card.

**noOfEventsPerType** is the number of events per type of event the card can record.

**noOfFaultsPerType** is the number of faults per type of fault the card can record.

**activityStructureLength** indicates the number of bytes available for storing activity records.

**noOfCardVehicleRecords** is the number of vehicle records the card can contain.

**noOfCardPlaceRecords** is the number of places the card can record.

Generation 2:

```
DriverCardApplicationIdentification ::= SEQUENCE {  
    typeOfTachographCardId      EquipmentType,  
    cardStructureVersion         CardStructureVersion,  
    noOfEventsPerType            NoOfEventsPerType,  
    noOfFaultsPerType           NoOfFaultsPerType,  
    activityStructureLength      CardActivityLengthRange,  
    noOfCardVehicleRecords      NoOfCardVehicleRecords,  
    noOfCardPlaceRecords        NoOfCardPlaceRecords,  
    noOfGNSSCDRecords           NoOfGNSSCDRecords,  
    noOfSpecificConditionRecords NoOfSpecificConditionRecords  
}
```

In addition to generation 1 the following data elements are used:

**noOfGNSSCDRecords** is the number of GNSS continuous driving records the card can store.

**noOfSpecificConditionRecords** is the number of specific condition records the card can store.

#### 2.62. DriverCardHolderIdentification

Information, stored in a driver card, related to the identification of the cardholder (Annex 1C requirement 256 and 281).

```
DriverCardHolderIdentification ::= SEQUENCE {  
    cardHolderName      HolderName,  
    cardHolderBirthDate Datef,  
    cardHolderPreferredLanguage Language  
}
```

**cardHolderName** is the name and first name(s) of the holder of the Driver Card.

**cardHolderBirthDate** is the date of birth of the holder of the Driver Card.

**cardHolderPreferredLanguage** is the preferred language of the card holder.

#### 2.63. DSRCSecurityData

Generation 2:

The plain text information and the MAC to be transmitted via DSRC from the tachograph to the Remote Interrogator (RI), see Appendix 11 Part B chapter 13 for details.

```

DSRCSecurityData ::= SEQUENCE {
    tagLenthPlainText          OCTET STRING(SIZE(2)),
    currentDateTime            CurrentDateTime,
    counter                    INTEGER(0..224-1),
    vuSerialNumber             VuSerialNumber,
    dSRCKMVersionNumber        INTEGER(SIZE(1)),
    tagLengthMac               OCTET STRING(SIZE(2)),
    mac                        MAC
}

```

**tagLength** is part of the DER-TLV encoding and shall be set to '81 10' (see Appendix 11 Part B chapter 13).

**currentDateTime** is the current date and time of the vehicle unit.

**counter** enumerates the RTM messages.

**vuSerialNumber** is the serial number of the vehicle unit.

**dSRCKMVersionNumber** is the version number of the DSRC Master Key from which the VU specific DSRC keys were derived.

**tagLengthMac** is the tag and length of the MAC data object as part of the DER-TLV encoding. The tag shall be set to '8E', the length shall encode the length of the MAC in octets (see Appendix 11 Part B chapter 13).

**mac** is the MAC calculated over the RTM message (see Appendix 11 Part B chapter 13).

#### 2.64. EGFCertificate

Generation 2:

Certificate of the external GNSS facility public key for mutual authentication with a VU. The structure of this certificate is specified in Appendix 11.

EGFCertificate ::= Certificate

#### 2.65. EmbedderIcAssemblerId

Provides information about the IC embedder.

```

EmbedderIcAssemblerId ::= SEQUENCE{
    countryCode                IA5String(SIZE(2)),
    moduleEmbedder             BCDString(SIZE(2)),
    manufacturerInformation     OCTET STRING(SIZE(1))
}

```

**countryCode** is the 2 letter country code of the module embedder according to ISO 3166.

**moduleEmbedder** identifies the module embedder.

**manufacturerInformation** for manufacturer internal usage.

## 2.66. EntryTypeDailyWorkPeriod

Code to distinguish between begin and end for an entry of a daily work period place and condition of the entry.

Generation 1

```
EntryTypeDailyWorkPeriod ::= INTEGER {  
    Begin,    related time = card insertion time or time of entry      (0),  
    End,      related time = card withdrawal time or time of entry    (1),  
    Begin,    related time manually entered (start time)              (2),  
    End,      related time manually entered (end of work period)      (3),  
    Begin,    related time assumed by VU                              (4),  
    End,      related time assumed by VU                              (5)  
}
```

**Value assignment:** according to ISO/IEC8824-1.

Generation 2

```
EntryTypeDailyWorkPeriod ::= INTEGER {  
    Begin,    related time = card insertion time or time of entry      (0),  
    End,      related time = card withdrawal time or time of entry    (1),  
    Begin,    related time manually entered (start time)              (2),  
    End,      related time manually entered (end of work period)      (3),  
    Begin,    related time assumed by VU                              (4),  
    End,      related time assumed by VU                              (5),  
    Begin,    related time based on GNSS data                        (6),  
    End,      related time based on GNSS data                        (7)  
}
```

**Value assignment:** according to ISO/IEC8824-1.

## 2.67. EquipmentType

Code to distinguish different types of equipment for the tachograph application.

EquipmentType ::= INTEGER(0..255)

## Generation 1:

--Reserved	(0),
--Driver Card	(1),
--Workshop Card	(2),
--Control Card	(3),
--Company Card	(4),
--Manufacturing Card	(5),
--Vehicle Unit	(6),
--Motion Sensor	(7),
--RFU	(8..255)

**Value assignment:** According to ISO/IEC8824-1.

Value 0 is reserved for the purpose of designating a Member State or Europe in the CHA field of certificates.

## Generation 2:

The same values as in generation 1 are used with the following additions:

--GNSS Facility	(8),
--Remote Communication Module	(9),
--ITS interface module	(10),
--Plaque	(11), -- may be used in SealRecord
--M1/N1 Adapter	(12), -- may be used in SealRecord
--European Root CA (ERCA)	(13),
--Member State CA (MSCA)	(14),
--External GNSS connection	(15), -- may be used in SealRecord
--Unused	(16), -- used in SealDataVu
--RFU	(17..255)

Note: The generation 2 values for the Plaque, Adapter and the External GNSS connection as well as the generation 1 values for the Vehicle Unit and Motion Sensor may be used in SealRecord, i.e. if applicable.

2.68. EuropeanPublicKey

Generation 1:

The European public key.

EuropeanPublicKey ::= PublicKey



## 2.69. EventFaultRecordPurpose

Code explaining why an event or a fault has been recorded.

EventFaultRecordPurpose ::= OCTET STRING (SIZE(1))

**Value assignment:**

'00'H	one of the 10 most recent (or last) events or faults
'01'H	the longest event for one of the last 10 days of occurrence
'02'H	one of the 5 longest events over the last 365 days
'03'H	the last event for one of the last 10 days of occurrence
'04'H	the most serious event for one of the last 10 days of occurrence
'05'H	one of the 5 most serious events over the last 365 days
'06'H	the first event or fault having occurred after the last calibration
'07'H	an active/on-going event or fault
'08'H to '7F'H	RFU
'80'H to 'FF'H	manufacturer specific

## 2.70. EventFaultType

Code qualifying an event or a fault.

EventFaultType ::= OCTET STRING (SIZE(1))

**Value assignment:**

## Generation 1:

'0x'H	General events,
'00'H	No further details,
'01'H	Insertion of a non valid card,
'02'H	Card conflict,
'03'H	Time overlap,
'04'H	Driving without an appropriate card,
'05'H	Card insertion while driving,
'06'H	Last card session not correctly closed,
'07'H	Over speeding,
'08'H	Power supply interruption,
'09'H	Motion data error,
'0A'H	Vehicle Motion Conflict,
'0B' to '0F'H	RFU,

'1x'H	Vehicle unit related security breach attempt events,
'10'H	No further details,
'11'H	Motion sensor authentication failure,
'12'H	Tachograph card authentication failure,
'13'H	Unauthorised change of motion sensor,
'14'H	Card data input integrity error
'15'H	Stored user data integrity error,
'16'H	Internal data transfer error,
'17'H	Unauthorised case opening,
'18'H	Hardware sabotage,
'19'H to '1F'H	RFU,
'2x'H	Sensor related security breach attempt events,
'20'H	No further details,
'21'H	Authentication failure,
'22'H	Stored data integrity error,
'23'H	Internal data transfer error,
'24'H	Unauthorised case opening,
'25'H	Hardware sabotage,
'26'H to '2F'H	RFU,
'3x'H	Recording equipment faults,
'30'H	No further details,
'31'H	VU internal fault,
'32'H	Printer fault,
'33'H	Display fault,
'34'H	Downloading fault,
'35'H	Sensor fault,
'36'H to '3F'H	RFU,
'4x'H	Card faults,
'40'H	No further details,
'41'H to '4F'H	RFU,
'50'H to '7F'H	RFU,
'80'H to 'FF'H	Manufacturer specific.

Generation 2:

The same values as in generation 1 are used with the following additions:

\0B'H	Time conflict (GNSS versus VU internal clock),
\0C' to '0F'H	RFU,
\5x'H	GNSS related faults,
'50'H	No further details,
'51'H	Internal GNSS receiver fault,
'52'H	External GNSS receiver fault,
'53'H	External GNSS communication fault,
'54'H	No GNSS position data,
'55'H	Tamper detection of GNSS,
'56'H	External GNSS facility certificate expired,
'57'H to '5F'H	RFU,
'6x'H	Remote Communication Module related faults,
'60'H	No further details,
'61'H	Remote Communication Module fault,
'62'H	Remote Communication Module communication fault,
'63'H to '6F'H	RFU,
'7x'H	ITS interface faults,
'70'H	No further details,
'71'H to '7F'H	RFU.

## 2.71. ExtendedSealIdentifier

Generation 2:

The extended seal identifier uniquely identifies a seal (Annex 1C requirement 401).

```
ExtendedSealIdentifier ::= SEQUENCE{
    manufacturerCode          OCTET STRING (SIZE(2)),
    sealIdentifier            OCTET STRING (SIZE(6) )
}
```

**manufacturerCode** is a code of the manufacturer of the seal.

**sealIdentifier** is an identifier for the seal which is unique for the manufacturer.

## 2.72. ExtendedSerialNumber

Unique identification of an equipment. It can also be used as an equipment Public Key Identifier.

Generation 1:

```
ExtendedSerialNumber ::= SEQUENCE{
    serialNumber              INTEGER(0..232-1),
    monthYear                 BCDString(SIZE(2)),
    type                       OCTET STRING(SIZE(1)),
    manufacturerCode          ManufacturerCode
}
```

**serialNumber** is a serial number for the equipment, unique for the manufacturer, the equipment's type and the month and year below.

**monthYear** is the identification of the month and the year of manufacturing (or of serial number assignment).

**Value assignment:** BCD coding of Month (two digits) and Year (two last digits).

**type** is an identifier of the type of equipment.

**Value assignment:** manufacturer specific, with 'FFh' reserved value.

**manufacturerCode:** is the numerical code identifying a manufacturer of type approved equipment.

Generation 2:

```
ExtendedSerialNumber ::= SEQUENCE{
    serialNumber              INTEGER(0..232-1),
    monthYear                 BCDString(SIZE(2)),
    type                       EquipmentType,
    manufacturerCode          ManufacturerCode
}
```

**serialNumber** see Generation 1

**monthYear** see Generation 1

**type** indicates the type of equipment

**manufacturerCode:** see Generation 1.

## 2.73. FullCardNumber

Code fully identifying a tachograph card.

```
FullCardNumber ::= SEQUENCE {
    cardType                EquipmentType,
    cardIssuingMemberState  NationNumeric,
    cardNumber              CardNumber
}
```

**cardType** is the type of the tachograph card.

**cardIssuingMemberState** is the code of the Member State having issued the card.

**cardNumber** is the card number.

## 2.74. FullCardNumberAndGeneration

Generation 2:

Code fully identifying a tachograph card and its generation.

```
FullCardNumberAndGeneration ::= SEQUENCE {
    fullCardNumber          FullCardNumber,
    generation              Generation
}
```

**fullcardNumber** identifies the tachograph card.

**generation** indicates the generation of the tachograph card used.

## 2.75. Generation

Generation 2:

Indicates the generation of tachograph used.

Generation ::= INTEGER(0..255)

**Value assignment:**

‘00’H      RFU

‘01’H      Generation 1

‘02’H      Generation 2

‘03’H .. ‘FF’H RFU

## 2.76. GeoCoordinates

Generation 2:

The geo-coordinates are encoded as integers. These integers are multiples of the  $\pm$ DDMM.M encoding for the latitude and  $\pm$ DDDMM.M for the longitude. Here  $\pm$ DD respectively  $\pm$ DDD denotes the degrees and MM.M the minutes.

```
GeoCoordinates ::= SEQUENCE {  
    latitude          INTEGER(-90000..90001),  
    longitude         INTEGER(-180000..180001)  
}
```

**latitude** is encoded as a multiple (factor 10) of the  $\pm$ DDMM.M representation.

**longitude** is encoded as a multiple (factor 10) of the  $\pm$ DDDMM.M representation.

#### 2.77. GNSSAccuracy

Generation 2:

The accuracy of the GNSS position data (definition eee)). This accuracy is encoded as integer and is a multiple (factor 10) of the X.Y value provided by the GSA NMEA sentence.

```
GNSSAccuracy ::= INTEGER(1..100)
```

#### 2.78. GNSSContinuousDriving

Generation 2:

Information, stored in a driver or workshop card, related to the GNSS position of the vehicle if the continuous driving time of the driver reaches a multiple of three hours (Annex 1C requirement 306 and 354).

```
GNSSContinuousDriving ::= SEQUENCE {  
    gnssCDPointerNewestRecord    INTEGER(0..NoOfGNSSCDRecords -1),  
    gnssContinuousDrivingRecords SET SIZE(NoOfGNSSCDRecords) OF  
                                GNSSContinuousDrivingRecord  
}
```

**gnssCDPointerNewestRecord** is the index of the last updated GNSS continuous driving record.

**Value assignment:** Number corresponding to the numerator of the GNSS continuous driving record, beginning with '0' for the first occurrence of the GNSS continuous driving record in the structure.

**gnssContinuousDrivingRecords** is the set of records containing the date and time the continuous driving reaches a multiple of three hours and information on the position of the vehicle.

#### 2.79. GNSSContinuousDrivingRecord

Generation 2:

Information, stored in a driver or workshop card, related to the GNSS position of the vehicle if the continuous driving time of the driver reaches a multiple of three hours (Annex 1C requirement 305 and 353).

GNSSContinuousDrivingRecord ::= SEQUENCE {

timeStamp	TimeReal,
gnssPlaceRecord	GNSSPlaceRecord

}

**timeStamp** is the date and time when the continuous driving time of the card holder reaches a multiple of three hours.

**gnssPlaceRecord** contains information related to the position of the vehicle.

#### 2.80. GNSSPlaceRecord

Generation 2:

Information related to the GNSS position of the vehicle (Annex 1C requirements 108, 109, 110, 296, 305, 347, and 353).

GNSSPlaceRecord ::= SEQUENCE {

timeStamp	TimeReal,
gnssAccuracy	GNSSAccuracy,
geoCoordinates	GeoCoordinates

}

**timeStamp** is the date and time when the GNSS position of the vehicle was determined.

**gnssAccuracy** is the accuracy of the GNSS position data.

**geoCoordinates** is the recorded location using GNSS.

#### 2.81. HighResOdometer

Odometer value of the vehicle: Accumulated distance travelled by the vehicle during its operation.

HighResOdometer ::= INTEGER(0..2<sup>32</sup>-1)

**Value assignment:** Unsigned binary. Value in 1/200 km in the operating range 0 to 21 055 406 km.

#### 2.82. HighResTripDistance

A distance travelled during all or part of a journey.

HighResTripDistance ::= INTEGER(0..2<sup>32</sup>-1)

**Value assignment:** Unsigned binary. Value in 1/200 km in the operating range 0 to 21 055 406 km.

## 2.83. HolderName

The surname and first name(s) of a card holder.

```
HolderName ::= SEQUENCE {  
    holderSurname                Name,  
    holderFirstNames            Name  
}
```

**holderSurname** is the surname (family name) of the holder. This surname does not include titles.

**Value assignment:** When a card is not personal, holderSurname contains the same information as companyName or workshopName or controlBodyName.

**holderFirstNames** is the first name(s) and initials of the holder.

## 2.84. InternalGNSSReceiver

Generation 2:

Information if the GNSS receiver is internal or external to the vehicle unit. True means that the GNSS receiver is internal to the VU. False means that the GNSS receiver is external.

```
InternalGNSSReceiver ::= BOOLEAN
```

## 2.85. K-ConstantOfRecordingEquipment

Constant of the recording equipment (definition m)).

```
K-ConstantOfRecordingEquipment ::= INTEGER(0..216-1)
```

**Value assignment:** Pulses per kilometer in the operating range 0 to 64 255 pulses/km.

## 2.86. KeyIdentifier

A unique identifier of a Public Key used to reference and select the key. It also identifies the holder of the key.

```
KeyIdentifier ::= CHOICE {  
    extendedSerialNumber        ExtendedSerialNumber,  
    certificateRequestID        CertificateRequestID,  
    certificationAuthorityKID    CertificationAuthorityKID  
}
```

The first choice is suitable to reference the public key of a Vehicle Unit or of a tachograph card.

The second choice is suitable to reference the public key of a Vehicle Unit (in the case the serial number of the Vehicle Unit cannot be known at certificate generation time).

The third choice is suitable to reference the public key of a Member State.

## 2.87. KMWCKey

Generation 2:

AES key and its associated key version used for VU – Motion Sensor pairing. For details see Appendix 11.



```

KMWCKey ::= SEQUENCE {
    kMWCKey          AESKey,
    keyVersion       INTEGER (SIZE(1))
}

```

**KMWCKey** is the length of the AES key concatenated with the key which is used for VU – Motion Sensor pairing.

**keyVersion** denotes the key version of the AES key.

#### 2.88. Language

Code identifying a language.

```
Language ::= IA5String(SIZE(2))
```

**Value assignment:** Two-letter lower-case coding according to ISO 639.

#### 2.89. LastCardDownload

Date and time, stored on a driver card, of last card download (for other purposes than control) Annex 1C requirement 257 and 282. This date is updateable by a VU or any card reader.

```
LastCardDownload ::= TimeReal
```

**Value assignment:** not further specified.

#### 2.90. LinkCertificate

Generation 2:

The link certificate between European Root CA key pairs.

```
LinkCertificate ::= Certificate
```

#### 2.91. L-TyreCircumference

Effective circumference of the wheel tyres (definition u).

```
L-TyreCircumference ::= INTEGER(0.. 216-1)
```

**Value assignment:** Unsigned binary, value in 1/8 mm in the operating range 0 to 8 031 mm.

#### 2.92. MAC

Generation 2:

A cryptographic checksum of 8, 12 or 16 bytes length corresponding to the cipher suites specified in Appendix 11.

```

MAC ::= CHOICE {
    mac8          OCTET STRING (SIZE(8)),
    mac12         OCTET STRING (SIZE(12)),
    mac16         OCTET STRING (SIZE(12))
}

```

## 2.93. ManualInputFlag

Code identifying whether a cardholder has manually entered driver activities at card insertion or not (Annex 1B requirement 081 and Annex 1C requirement 102).

```
ManualInputFlag ::= INTEGER {  
    noEntry                (0)  
    manualEntries          (1)  
}
```

**Value assignment:** not further specified.

## 2.94. ManufacturerCode

Code identifying a manufacturer of type approved equipment.

```
ManufacturerCode ::= INTEGER(0..255)
```

The laboratory competent for interoperability tests maintains and publishes the list of manufacturer codes on its web site (Annex 1C requirement 454).

ManufacturerCodes are provisionally assigned to developers of tachograph equipment on application to the laboratory competent for interoperability tests.

## 2.95. ManufacturerSpecificEventFaultData

Generation 2:

Manufacturer specific error codes simplify the error analysis and maintenance of vehicle units.

```
ManufacturerSpecificEventFaultData ::= SEQUENCE {  
    manufacturerCode          ManufacturerCode,  
    manufacturerSpecificErrorCode  OCTET STRING(SIZE(3))  
}
```

**manufacturerCode** identifies the manufacturer of the Vehicle Unit.

**manufacturerSpecificErrorCode** is an error code specific to the manufacturer.

## 2.96. MemberStateCertificate

The certificate of the public key of a member state issued by the European certification authority.

```
MemberStateCertificate ::= Certificate
```

## 2.97. MemberStateCertificateRecordArray

Generation 2:

The member state certificate plus metadata as used in the download protocol.

```

MemberStateCertificateRecordArray ::= SEQUENCE {
    recordType          RecordType,
    recordSize          INTEGER(1..65535),
    noOfRecords         INTEGER(0..65535),
    records             SET SIZE(noOfRecords) OF MemberStateCertificate
}

```

**recordType** denotes the type of the record (MemberStateCertificate). **Value Assignment:** See RecordType

**recordSize** is the size of the MemberStateCertificate in bytes.

**noOfRecords** is the number of records in the set records. The value shall be set to 1 as the certificates may have different lengths.

**records** is the set of member state certificates.

#### 2.98. MemberStatePublicKey

Generation 1:

The public key of a Member State.

MemberStatePublicKey ::= PublicKey

#### 2.99. Name

A name.

```

Name ::= SEQUENCE {
    codePage            INTEGER (0..255),
    name                OCTET STRING (SIZE(35))
}

```

**codePage** specifies a character set defined in Chapter 4,

**name** is a name encoded using the specified character set.

#### 2.100. NationAlpha

Alphabetic reference to a country shall be in accordance with the distinguishing signs used on vehicles in international traffic (United Nations Vienna Convention on Road Traffic, 1968).

NationAlpha ::= IA5String(SIZE(3))

The Nation Alpha and Numeric codes shall be held on a list maintained on the website of the laboratory appointed to carry out interoperability testing, as set out in Annex 1C requirement 440.

#### 2.101. NationNumeric

Numerical reference to a country.

NationNumeric ::= INTEGER(0 .. 255)

**Value assignment:** see data type 2.100 (NationAlpha).

Any amendment or updating of the Nation Alpha or Numeric specification described in the above paragraph shall only be made out after the appointed laboratory has obtained the views of type approved digital and smart tachograph vehicle unit manufacturers.

#### 2.102. NoOfCalibrationRecords

Number of calibration records, a workshop card can store.

Generation 1:

NoOfCalibrationRecords ::= INTEGER(0..255)

**Value assignment:** see Appendix 2.

Generation 2:

NoOfCalibrationRecords ::= INTEGER(0..2<sup>16</sup>-1)

**Value assignment:** see Appendix 2.

#### 2.103. NoOfCalibrationsSinceDownload

Counter indicating the number of calibrations performed with a workshop card since its last download (Annex 1C requirement 317 and 340).

NoOfCalibrationsSinceDownload ::= INTEGER(0..2<sup>16</sup>-1)

**Value assignment:** Not specified further.

#### 2.104. NoOfCardPlaceRecords

Number of place records a driver or workshop card can store.

Generation 1:

NoOfCardPlaceRecords ::= INTEGER(0..255)

**Value assignment:** see Appendix 2.

Generation 2:

NoOfCardPlaceRecords ::= INTEGER(0..2<sup>16</sup>-1)

**Value assignment:** see Appendix 2.

## 2.105. NoOfCardVehicleRecords

Number of vehicles used records a driver or workshop card can store.

NoOfCardVehicleRecords ::= INTEGER(0..  $2^{16}-1$ )

**Value assignment:** see Appendix 2.

## 2.106. NoOfCardVehicleUnitRecords

Generation 2:

Number of vehicle units used records a driver or workshop card can store.

NoOfCardVehicleUnitRecords ::= INTEGER(0..  $2^{16}-1$ )

**Value assignment:** see Appendix 2.

## 2.107. NoOfCompanyActivityRecords

Number of company activity records, a company card can store.

NoOfCompanyActivityRecords ::= INTEGER(0..  $2^{16}-1$ )

**Value assignment:** see Appendix 2.

## 2.108. NoOfControlActivityRecords

Number of control activity records, a control card can store.

NoOfControlActivityRecords ::= INTEGER(0..  $2^{16}-1$ )

**Value assignment:** see Appendix 2.

## 2.109. NoOfEventsPerType

Number of events per type of event a card can store.

NoOfEventsPerType ::= INTEGER(0..255)

**Value assignment:** see Appendix 2.

## 2.110. NoOfFaultsPerType

Number of faults per type of fault a card can store.

NoOfFaultsPerType ::= INTEGER(0..255)

**Value assignment:** see Appendix 2.

## 2.111. NoOfGNSSCDRecords

Generation 2:

Number of GNSS continuous driving records a card can store.

NoOfGNSSCDRecords ::= INTEGER(0.. $2^{16}-1$ )

**Value assignment:** see Appendix 2.

## 2.112. NoOfSpecificConditionRecords

Generation 2:

Number of specific condition records a card can store.

NoOfSpecificConditionRecords ::= INTEGER(0..2<sup>16</sup>-1)

**Value assignment:** see Appendix 2.

## 2.113. OdometerShort

Odometer value of the vehicle in a short form.

OdometerShort ::= INTEGER(0..2<sup>24</sup>-1)

**Value assignment:** Unsigned binary. Value in km in the operating range 0 to 9 999 999 km.

## 2.114. OdometerValueMidnight

The vehicle's odometer value at midnight on a given day (Annex 1B requirement 090 and Annex 1C requirement 113).

OdometerValueMidnight ::= OdometerShort

**Value assignment:** not further specified.

## 2.115. OdometerValueMidnightRecordArray

Generation 2:

The OdometerValueMidnight plus metadata used in the download protocol.

OdometerValueMidnightRecordArray ::= SEQUENCE {

recordType	RecordType,
recordSize	INTEGER(1..65535),
noOfRecords	INTEGER(0..65535),
records	SET SIZE(noOfRecords) OF OdometerValueMidnight

}

**recordType** denotes the type of the record (OdometerValueMidnight). **Value Assignment:** See RecordType

**recordSize** is the size of the OdometerValueMidnight in bytes.

**noOfRecords** is the number of records in the set records.

**records** is the set of OdometerValueMidnight records.

## 2.116. OverspeedNumber

Number of over speeding events since the last over speeding control.

OverspeedNumber ::= INTEGER(0..255)

**Value assignment:** 0 means that no over speeding event has occurred since the last over speeding control , 1 means that one over speeding event has occurred since the last over speeding control ...255 means that 255 or more over speeding events have occurred since the last over speeding control.

## 2.117. PlaceRecord

Information related to a place where a daily work period begins or ends (Annex 1C requirements 108, 271, 296, 324, and 347).

Generation 1:

```
PlaceRecord ::= SEQUENCE {
    entryTime                TimeReal,
    entryTypeDailyWorkPeriod EntryTypeDailyWorkPeriod,
    dailyWorkPeriodCountry   NationNumeric,
    dailyWorkPeriodRegion    RegionNumeric,
    vehicleOdometerValue     OdometerShort
}
```

**entryTime** is a date and time related to the entry.

**entryTypeDailyWorkPeriod** is the type of entry.

**dailyWorkPeriodCountry** is the country entered.

**dailyWorkPeriodRegion** is the region entered.

**vehicleOdometerValue** is the odometer value at the time of place entry.

Generation 2:

```
PlaceRecord ::= SEQUENCE {  
    entryTime                TimeReal,  
    entryTypeDailyWorkPeriod EntryTypeDailyWorkPeriod,  
    dailyWorkPeriodCountry   NationNumeric,  
    dailyWorkPeriodRegion    RegionNumeric,  
    vehicleOdometerValue     OdometerShort,  
    entryGNSSPlaceRecord     GNSSPlaceRecord  
}
```

In addition to Generation 1 the following component is used:

**entryGNSSPlaceRecord** is the recorded location and time.

#### 2.118. PreviousVehicleInfo

Information related to the vehicle previously used by a driver when inserting his card in a vehicle unit (Annex 1B requirement 081 and Annex 1C requirement 102).

Generation 1:

```
PreviousVehicleInfo ::= SEQUENCE {  
    vehicleRegistrationIdentification VehicleRegistrationIdentification,  
    cardWithdrawalTime              TimeReal  
}
```

**vehicleRegistrationIdentification** is the VRN and the registering Member State of the vehicle.

**cardWithdrawalTime** is the card withdrawal date and time.

Generation 2:

```
PreviousVehicleInfo ::= SEQUENCE {  
    vehicleRegistrationIdentification VehicleRegistrationIdentification,  
    cardWithdrawalTime              TimeReal,  
    vuGeneration                    Generation  
}
```

In addition to generation 1 the following data element is used:

**vuGeneration** identifies the generation of the vehicle unit.



## 2.119. PublicKey

Generation 1:

A public RSA key.

```
PublicKey ::= SEQUENCE {
    rsaKeyModulus          RSAKeyModulus,
    rsaKeyPublicExponent  RSAKeyPublicExponent
}
```

**rsaKeyModulus** is the Modulus of the key pair.

**rsaKeyPublicExponent** is the public exponent of the key pair.

## 2.120. RecordType

Generation 2:

Reference to a record type. This data type is used in RecordArrays.

```
RecordType ::= OCTET STRING(SIZE(1))
```

**Value assignment:**

'01'H	ActivityChangeInfo,
'02'H	CardSlotsStatus,
'03'H	CurrentDateTime,
'04'H	MemberStateCertificate,
'05'H	OdometerValueMidnight,
'06'H	DateOfDayDownloaded,
'07'H	SensorPaired,
'08'H	Signature,
'09'H	SpecificConditionRecord,
'0A'H	VehicleIdentificationNumber,
'0B'H	VehicleRegistrationNumber,
'0C'H	VuCalibrationRecord,
'0D'H	VuCardIWRecord,
'0E'H	VuCardRecord,
'0F'H	VuCertificate,
'10'H	VuCompanyLocksRecord,
'11'H	VuControlActivityRecord,
'12'H	VuDetailedSpeedBlock,
'13'H	VuDownloadablePeriod,

'14'H	VuDownloadActivityData,
'15'H	VuEventRecord,
'16'H	VuGNSSCDRecord,
'17'H	VuTSCconsentRecord,
'18'H	VuFaultRecord,
'19'H	VuIdentification,
'1A'H	VuOverSpeedingControlData,
'1B'H	VuOverSpeedingEventRecord,
'1C'H	VuPlaceDailyWorkPeriodRecord,
'1D'H	VuTimeAdjustmentGNSSRecord,
'1E'H	VuTimeAdjustmentRecord,
'1F'H	VuPowerSupplyInterruptionRecord,
'20'H	SensorPairedRecord,
'21'H	SensorExternalGNSSCoupledRecord,
'22'H to '7F'H	RFU,
'80'H to 'FF'H	Manufacturer specific.

#### 2.121. RegionAlpha

Alphabetic reference to a region within a specified country.

RegionAlpha ::= IA5STRING(SIZE(3))

Generation 1:

#### Value assignment:

' '	No information available,
Spain:	
'AN'	Andalucía,
'AR'	Aragón,
'AST'	Asturias,
'C'	Cantabria,
'CAT'	Cataluña,
'CL'	Castilla-León,
'CM'	Castilla-La-Mancha,
'CV'	Valencia,
'EXT'	Extremadura,

'G '	Galicia,
'IB '	Baleares,
'IC '	Canarias,
'LR '	La Rioja,
'M '	Madrid,
'MU '	Murcia,
'NA '	Navarra,
'PV '	País Vasco

Generation 2:

The RegionAlpha codes shall be held on a list maintained on the website of the laboratory appointed to carry out interoperability testing.

## 2.122. RegionNumeric

Numerical reference to a region within a specified country.

RegionNumeric ::= OCTET STRING (SIZE(1))

Generation 1:

**Value assignment:**

'00'H No information available,

Spain:

'01'H Andalucía,

'02'H Aragón,

'03'H Asturias,

'04'H Cantabria,

'05'H Cataluña,

'06'H Castilla-León,

'07'H Castilla-La-Mancha,

'08'H Valencia,

'09'H Extremadura,

'0A'H Galicia,

'0B'H Baleares,

'0C'H Canarias,

'0D'H La Rioja,

'0E'H Madrid,

'0F'H Murcia,

'10'H Navarra,

'11'H País Vasco

Generation 2:

The RegionNumeric codes shall be held on a list maintained on the website of the laboratory appointed to carry out interoperability testing.

## 2.123. RemoteCommunicationModuleSerialNumber

Generation 2:

Serial number of the Remote Communication Module.

RemoteCommunicationModuleSerialNumber ::= ExtendedSerialNumber

## 2.124. RSAKeyModulus

Generation 1:

The modulus of a RSA key pair.

RSAKeyModulus ::= OCTET STRING (SIZE(128))

**Value assignment:** Unspecified.

## 2.125. RSAKeyPrivateExponent

Generation 1:

The private exponent of a RSA key pair.

RSAKeyPrivateExponent ::= OCTET STRING (SIZE(128))

**Value assignment:** Unspecified.

## 2.126. RSAKeyPublicExponent

Generation 1:

The public exponent of a RSA key pair.

RSAKeyPublicExponent ::= OCTET STRING (SIZE(8))

**Value assignment:** Unspecified.

## 2.127. RtmData

Generation 2:

For the definition of this data type see Appendix 14.

## 2.128. SealDataCard

Generation 2:

This data type stores information about the seals that are attached to the different components of a vehicle and is intended for storage on a card. This data type is related to Annex 1C requirement 337.

SealDataCard ::= SEQUENCE {

noOfSealRecords INTEGER(1..5),

sealRecords SET SIZE(noOfSealRecords) OF SealRecord

}

**noOfSealRecords** is the number of records in sealRecords.

**sealRecords** is a set of seal records.

## 2.129. SealDataVu

Generation 2:

This data type stores information about the seals that are attached to the different components of a vehicle and is intended for storage in a Vehicle Unit.

SealDataVu ::= SEQUENCE SIZE(5) OF {

sealRecords	SealRecord
-------------	------------

}

**sealRecords** is a set of seal records. If there are less than 5 seals available the value of the EquipmentType in all unused sealRecords shall be set to 16, i.e. unused.

## 2.130. SealRecord

Generation 2:

This data type stores information about a seal that is attached to a component. This data type is related to Annex 1C requirement 337.

SealRecord ::= SEQUENCE {

equipmentType	EquipmentType,
extendedSealIdentifier	ExtendedSealIdentifier

}

**equipmentType** identifies the type of equipment the seal is attached to.

**extendedSealIdentifier** is the identifier of the seal attached to the equipment.

## 2.131. SensorApprovalNumber

Type approval number of the sensor.

Generation 1:

SensorApprovalNumber ::= IA5String(SIZE(8))

**Value assignment:** Unspecified.

Generation 2:

SensorApprovalNumber ::= IA5String(SIZE(16))

**Value assignment:**

The approval number shall be provided as published on the corresponding European Commission web site, i.e. for example including hyphens if any. The approval number shall be left-aligned.

## 2.132. SensorExternalGNSSApprovalNumber

Generation 2:

Type approval number of the external GNSS facility.

SensorExternalGNSSApprovalNumber ::= IA5String(SIZE(16))

**Value assignment:**

The approval number shall be provided as published on the corresponding European Commission web site, i.e. for example including hyphens if any. The approval number shall be left-aligned.

#### 2.133. SensorExternalGNSSCoupledRecord

Generation 2:

Information, stored in a vehicle unit, related to the identification of the external GNSS facility coupled with the vehicle unit (Annex 1C requirement 100).

SensorExternalGNSSCoupledRecord ::= SEQUENCE {

sensorSerialNumber	SensorGNSSSerialNumber,
sensorApprovalNumber	SensorExternalGNSSApprovalNumber,
sensorCouplingDate	SensorGNSSCouplingDate

}

**sensorSerialNumber** is the serial number of the external GNSS facility coupled with the vehicle unit.

**sensorApprovalNumber** is the approval number of this external GNSS facility.

**sensorCouplingDate** is a date of coupling of this external GNSS facility with the vehicle unit.

## 2.134. SensorExternalGNSSIdentification

Generation 2:

Information related to the identification of the external GNSS facility (Annex 1C requirement 98).

SensorExternalGNSSIdentification ::= SEQUENCE {

sensorSerialNumber	SensorGNSSSerialNumber,
sensorApprovalNumber	SensorExternalGNSSApprovalNumber,
sensorSCIdentifier	SensorExternalGNSSSCIdentifier,
sensorOSIdentifier	SensorExternalGNSSOSIdentifier

}

**sensorSerialNumber** is the extended serial number of the external GNSS facility.

**sensorApprovalNumber** is the approval number of the external GNSS facility.

**sensorSCIdentifier** is the identifier of the security component of the external GNSS facility.

**sensorOSIdentifier** is the identifier of the operating system of the external GNSS facility.

## 2.135. SensorExternalGNSSInstallation

Generation 2:

Information, stored in an external GNSS facility, related to the installation of the external GNSS sensor (Annex 1C requirement 123).

SensorExternalGNSSInstallation ::= SEQUENCE {

sensorCouplingDateFirst	SensorGNSSCouplingDate,
firstVuApprovalNumber	VuApprovalNumber,
firstVuSerialNumber	VuSerialNumber,
sensorCouplingDateCurrent	SensorGNSSCouplingDate,
currentVuApprovalNumber	VuApprovalNumber,
currentVUSerialNumber	VuSerialNumber

}

**sensorCouplingDateFirst** is the date of the first coupling of external GNSS facility with a vehicle unit.

**firstVuApprovalNumber** is the approval number of the first vehicle unit coupled with the external GNSS facility.

**firstVuSerialNumber** is the serial number of the first vehicle unit paired with the external GNSS facility.

**sensorCouplingDateCurrent** is the date of the current coupling of external GNSS facility with a vehicle unit.

**currentVuApprovalNumber** is the approval number of the vehicle unit currently coupled with the external GNSS facility.

**currentVUSerialNumber** is the serial number of the vehicle unit currently coupled with the external GNSS facility.



## 2.136. SensorExternalGNSSOSIdentifier

Generation 2:

Identifier of the operating system of the external GNSS facility.

SensorOSIdentifier ::= IA5String(SIZE(2))

**Value assignment:** manufacturer specific.

## 2.137. SensorExternalGNSSSecurityIdentifier

Generation 2:

This type is used e.g. to identify the cryptographic module of the external GNSS facility.

Identifier of the security component of the external GNSS facility.

SensorExternalGNSSSecurityIdentifier ::= IA5String(SIZE(8))

**Value assignment:** component manufacturer specific.

## 2.138. SensorGNSSCouplingDate

Generation 2:

Date of a coupling of the external GNSS facility with a vehicle unit.

SensorGNSSCouplingDate ::= TimeReal

**Value assignment:** Unspecified.

#### 2.139. SensorGNSSSerialNumber

Generation 2:

This type is used to store the serial number of the GNSS receiver both when it is inside the VU and when it is outside the VU.

Serial number of the GNSS receiver.

SensorGNSSSerialNumber ::= ExtendedSerialNumber

#### 2.140. SensorIdentification

Information, stored in a motion sensor, related to the identification of the motion sensor (Annex 1B requirement 077 and Annex 1C requirement 95).

SensorIdentification ::= SEQUENCE {

sensorSerialNumber	SensorSerialNumber,
sensorApprovalNumber	SensorApprovalNumber,
sensorSCIdentifier	SensorSCIdentifier,
sensorOSIdentifier	SensorOSIdentifier

}

**sensorSerialNumber** is the extended serial number of the motion sensor (includes part number and manufacturer code).

**sensorApprovalNumber** is the approval number of the motion sensor.

**sensorSCIdentifier** is the identifier of the security component of the motion sensor.

**sensorOSIdentifier** is the identifier of the operating system of the motion sensor.

## 2.141. SensorInstallation

Information, stored in a motion sensor, related to the installation of the motion sensor (Annex 1B requirement 099 and Annex 1C requirement 122).

SensorInstallation ::= SEQUENCE {

sensorPairingDateFirst	SensorPairingDate,
firstVuApprovalNumber	VuApprovalNumber,
firstVuSerialNumber	VuSerialNumber,
sensorPairingDateCurrent	SensorPairingDate,
currentVuApprovalNumber	VuApprovalNumber,
currentVUSerialNumber	VuSerialNumber

}

**sensorPairingDateFirst** is the date of the first pairing of the motion sensor with a vehicle unit.

**firstVuApprovalNumber** is the approval number of the first vehicle unit paired with the motion sensor.

**firstVuSerialNumber** is the serial number of the first vehicle unit paired with the motion sensor.

**sensorPairingDateCurrent** is the date of the current pairing of the motion sensor with the vehicle unit.

**currentVuApprovalNumber** is the approval number of the vehicle unit currently paired with the motion sensor.

**currentVUSerialNumber** is the serial number of the vehicle unit currently paired with the motion sensor.

## 2.142. SensorInstallationSecData

Information, stored in a workshop card, related to the security data needed for pairing motion sensors to vehicle units (Annex 1C requirement 308 and 331).

Generation 1:

SensorInstallationSecData ::= TDesSessionKey

**Value assignment:** in accordance with ISO 16844-3.

Generation 2:

As described in Appendix 11 a workshop card shall store up to three keys for VU Motion Sensor pairing. These keys have different key versions.

```
SensorInstallationSecData ::= SEQUENCE {  
    kMWCKey1          KMWCKey,  
    kMWCKey2          KMWCKey OPTIONAL,  
    kMWCKey3          KMWCKey OPTIONAL  
}
```

#### 2.143. SensorOSIdentifier

Identifier of the operating system of the motion sensor.

SensorOSIdentifier ::= IA5String(SIZE(2))

**Value assignment:** manufacturer specific.

#### 2.144. SensorPaired

Generation 1:

Information, stored in a vehicle unit, related to the identification of the motion sensor paired with the vehicle unit (Annex 1B requirement 079).

```
SensorPaired ::= SEQUENCE {  
    sensorSerialNumber      SensorSerialNumber,  
    sensorApprovalNumber    SensorApprovalNumber,  
    sensorPairingDateFirst  SensorPairingDate  
}
```

**sensorSerialNumber** is the serial number of the motion sensor currently paired with the vehicle unit.

**sensorApprovalNumber** is the approval number of the motion sensor currently paired with the vehicle unit.

**sensorPairingDateFirst** is the date of the first pairing with a vehicle unit of the motion sensor currently paired with the vehicle unit.

#### 2.145. SensorPairedRecord

Generation 2:

Information, stored in a vehicle unit, related to the identification of a motion sensor paired with the vehicle unit (Annex 1C requirement 97).

```
SensorPairedRecord ::= SEQUENCE {  
    sensorSerialNumber      SensorSerialNumber,  
    sensorApprovalNumber    SensorApprovalNumber,  
    sensorPairingDate       SensorPairingDate  
}
```

**sensorSerialNumber** is the serial number of a motion sensor paired with the vehicle unit.

**sensorApprovalNumber** is the approval number of this motion sensor.

**sensorPairingDate** is a date of pairing of this motion sensor with the vehicle unit.

## 2.146. SensorPairingDate

Date of a pairing of the motion sensor with a vehicle unit.

SensorPairingDate ::= TimeReal

**Value assignment:** Unspecified.

## 2.147. SensorSCIdentifier

Identifier of the security component of the motion sensor.

SensorSCIdentifier ::= IA5String(SIZE(8))

**Value assignment:** component manufacturer specific.

## 2.148. SensorSerialNumber

Serial number of the motion sensor.

SensorSerialNumber ::= ExtendedSerialNumber

## 2.149. Signature

A digital signature.

Generation 1:

Signature ::= OCTET STRING (SIZE(128))

**Value assignment:** in accordance with Appendix 11 Common security mechanisms.

Generation 2:

Signature ::= OCTET STRING (SIZE(64..132))

**Value assignment:** in accordance with Appendix 11 Common security mechanisms.

## 2.150. SignatureRecordArray

Generation 2:

A set of signatures plus metadata used in the download protocol.

```
SignatureRecordArray ::= SEQUENCE {  
    recordType          RecordType,  
    recordSize          INTEGER(1..65535),  
    noOfRecords         INTEGER(0..65535),  
    records              SET SIZE(noOfRecords) OF Signature  
}
```

**recordType** denotes the type of the record (Signature). **Value Assignment:** See RecordType

**recordSize** is the size of the Signature in bytes.

**noOfRecords** is the number of records in the set records. The value shall be set to 1 as the signatures may have different lengths.

**records** is the set of signatures.

#### 2.151. SimilarEventsNumber

The number of similar events for one given day (Annex 1B requirement 094 and Annex 1C requirement 117).

SimilarEventsNumber ::= INTEGER(0..255)

**Value assignment:** 0 is not used, 1 means that only one event of that type has occurred and has been stored on that day, 2 means that 2 events of that type has occurred on that day (one only has been stored), ...255 means that 255 or more events of that type have occurred on that day.

#### 2.152. SpecificConditionRecord

Information, stored in a driver card, a workshop card or a vehicle unit, related to a specific condition (requirements Annex 1C 130, 276, 301, 328, and 355).

```
SpecificConditionRecord ::= SEQUENCE {  
    entryTime           TimeReal,  
    specificConditionType SpecificConditionType  
}
```

**entryTime** is the date and time of the entry.

**specificConditionType** is the code identifying the specific condition.

#### 2.153. SpecificConditions

Information, stored in a driver card, a workshop card or a vehicle unit, related to a specific condition (Annex 1C requirement 131, 277, 302, 329, and 356).

Generation 2:

```

SpecificConditions := SEQUENCE {
    conditionPointerNewestRecord          INTEGER(0..NoOfSpecificConditionRecords-1),
    specificConditionRecords              SET SIZE(NoOfSpecificConditionRecords) OF
                                          SpecificConditionRecord
}

```

**conditionPointerNewestRecord** is the index of the last updated specific condition record.

**Value assignment:** Number corresponding to the numerator of the specific condition record, beginning with '0' for the first occurrence of the specific condition record in the structure.

**specificConditionRecords** is the set of records containing information on the specific conditions recorded.

#### 2.154. SpecificConditionType

Code identifying a specific condition (Annex 1B requirements 050b, 105a, 212a and 230a and Annex 1C requirements 62).

SpecificConditionType ::= INTEGER(0..255)

Generation 1:

**Value assignment:**

'00'H	RFU
'01'H	Out of scope – Begin
'02'H	Out of scope – End
'03'H	Ferry / Train crossing
'04'H .. 'FF'H	RFU

Generation 2:

**Value assignment:**

'00'H	RFU
'01'H	Out of scope – Begin
'02'H	Out of scope – End
'03'H	Ferry / Train crossing – Begin
'04'H	Ferry / Train crossing – End
'05'H .. 'FF'H	RFU

#### 2.155. Speed

Speed of the vehicle (km/h).

Speed ::= INTEGER(0..255)

**Value assignment:** kilometers per hour in the operational range 0 to 220 km/h.

2.156. SpeedAuthorised

Maximum authorised Speed of the vehicle (definition hh)).

SpeedAuthorised ::= Speed

2.157. SpeedAverage

Average speed in a previously defined duration (km/h).

SpeedAverage ::= Speed

2.158. SpeedMax

Maximum speed measured in a previously defined duration.

SpeedMax ::= Speed

2.159. TachographPayload

Generation 2:

For the definition of this data type see Appendix 14.

2.160. TachographPayloadEncrypted

Generation 2:

The DER-TLV encrypted tachograph payload, i.e. the data sent encrypted in the RTM message. For the encryption see Appendix 11 Part B chapter 13.

TachographPayloadEncrypted ::= SEQUENCE {



tag	OCTET STRING(SIZE(1)),
length	OCTET STRING(SIZE(1..2)),
paddingContentIndicatorByte	OCTET STRING(SIZE(1)),
encryptedData	OCTET STRING(SIZE(16..192))
}	

**tag** is part of the DER-TLV encoding and shall be set to '87' (see Appendix 11 Part B chapter 13).

**length** is part of the DER-TLV encoding and shall encode the length of the following paddingContentIndicatorByte and the encryptedData.

**paddingContentIndicatorByte** shall be set to '00'.

**encryptedData** is encrypted tachographPayload as specified in Appendix 11 Part B chapter 13. The length of this data in octets shall always be a multiple of 16.

#### 2.161. TDesSessionKey

Generation 1:

A triple DES session key.

TDesSessionKey ::= SEQUENCE {

tDesKeyA	OCTET STRING (SIZE(8)),
tDesKeyB	OCTET STRING (SIZE(8))
}	

**Value assignment:** not further specified.

#### 2.162. TimeReal

Code for a combined date and time field, where the date and time are expressed as seconds past 00h.00m.00s. on 1 January 1970 GMT.

TimeReal{INTEGER:TimeRealRange} ::= INTEGER(0..TimeRealRange)

**Value assignment – Octet Aligned:** Number of seconds since midnight 1 January 1970 GMT.

The max. possible date/time is in the year 2106.

#### 2.163. TyreSize

Designation of tyre dimensions.

TyreSize ::= IA5String(SIZE(15))

**Value assignment:** in accordance with Directive 92/23 (EEC) 31/03/92 O.J. L129 p.95.

#### 2.164. VehicleIdentificationNumber

Vehicle Identification Number (VIN) referring to the vehicle as a whole, normally chassis serial number or frame number.

VehicleIdentificationNumber ::= IA5String(SIZE(17))

**Value assignment:** As defined in ISO 3779.

## 2.165. VehicleIdentificationNumberRecordArray

Generation 2:

The Vehicle Identification Number plus metadata as used in the download protocol.

```
VehicleIdentificationNumberRecordArray ::= SEQUENCE {  
    recordType                RecordType,  
    recordSize                INTEGER(1..65535),  
    noOfRecords              INTEGER(0..65535),  
    records                   SET SIZE(noOfRecords) OF VehicleIdentificationNumber  
}
```

**recordType** denotes the type of the record (VehicleIdentificationNumber). **Value Assignment:** See RecordType

**recordSize** is the size of the VehicleIdentificationNumber in bytes.

**noOfRecords** is the number of records in the set records.

**records** is the set of vehicle identification numbers.

## 2.166. VehicleRegistrationIdentification

Identification of a vehicle, unique for Europe (VRN and Member State).

```
VehicleRegistrationIdentification ::= SEQUENCE {  
    vehicleRegistrationNation  NationNumeric,  
    vehicleRegistrationNumber  VehicleRegistrationNumber  
}
```

**vehicleRegistrationNation** is the nation where the vehicle is registered.

**vehicleRegistrationNumber** is the registration number of the vehicle (VRN).

## 2.167. VehicleRegistrationNumber

Registration number of the vehicle (VRN). The registration number is assigned by the vehicle licensing authority.

```
VehicleRegistrationNumber ::= SEQUENCE {  
    codePage                  INTEGER (0..255),  
    vehicleRegNumber          OCTET STRING (SIZE(13))  
}
```

**codePage** specifies a character set defined in Chapter 4,

**vehicleRegNumber** is a VRN encoded using the specified character set.

**Value assignment:** Country specific.

## 2.168. VehicleRegistrationNumberRecordArray

Generation 2:

The Vehicle Registration Number plus metadata as used in the download protocol.

```
VehicleRegistrationNumberRecordArray ::= SEQUENCE {
    recordType          RecordType,
    recordSize          INTEGER(1..65535),
    noOfRecords         INTEGER(0..65535),
    records              SET SIZE(noOfRecords) OF VehicleRegistrationNumber
}
```

**recordType** denotes the type of the record (VehicleRegistrationNumber). **Value Assignment:** See RecordType

**recordSize** is the size of the VehicleRegistrationNumber in bytes.

**noOfRecords** is the number of records in the set records.

**records** is the set of vehicle registration numbers.

#### 2.169. VuAbility

Generation 2:

Information stored in a VU on the ability of the VU to use generation 1 tachograph cards or not (Annex 1C requirement 121).

VuAbility ::= OCTET STRING (SIZE(1))

**Value assignment – Octet Aligned :** ‘xxxxxxxa’B (8 bits)

For the ability to support of generation 1:

‘a’B            Ability to support generation 1 tachograph cards:  
                   ‘0’ B Generation 1 is supported,  
                   ‘1’ B Generation1 is not supported,  
                   ‘xxxxxxx’B    RFU

#### 2.170. VuActivityDailyData

Generation 1:

Information, stored in a VU, related to changes of activity and/or changes of driving status and/or changes of card status for a given calendar day (Annex 1B requirement 084 and Annex 1C requirement 105, 106, 107) and to slots status at 00:00 that day.

```
VuActivityDailyData ::= SEQUENCE {
    noOfActivityChanges            INTEGER SIZE(0..1440),
    activityChangeInfos            SET SIZE(noOfActivityChanges) OF ActivityChangeInfo
}
```

**noOfActivityChanges** is the number of ActivityChangeInfo words in the activityChangeInfos set.

**activityChangeInfos** is the set of ActivityChangeInfo words stored in the VU for the day. It always includes two ActivityChangeInfo words giving the status of the two slots at 00:00 that day.

## 2.171. VuActivityDailyRecordArray

Generation 2:

Information, stored in a VU, related to changes of activity and/or changes of driving status and/or changes of card status for a given calendar day (Annex 1C requirement 105, 106, 107) and to slots status at 00:00 that day.

VuActivityDailyRecordArray ::= SEQUENCE {

recordType	RecordType,
recordSize	INTEGER(1..65535),
noOfRecords	INTEGER(0..65535),
records	SET SIZE(noOfRecords) OF ActivityChangeInfo

}

**recordType** denotes the type of the record (ActivityChangeInfo). **Value Assignment:** See RecordType

**recordSize** is the size of the ActivityChangeInfo in bytes.

**noOfRecords** is the number of records in the set records.

**records** is the set of ActivityChangeInfo words stored in the VU for the day. It always includes two ActivityChangeInfo words giving the status of the two slots at 00:00 that day.

## 2.172. VuApprovalNumber

Type approval number of the vehicle unit.

Generation 1:

VuApprovalNumber ::= IA5String(SIZE(8))

**Value assignment:** Unspecified.

Generation 2:

VuApprovalNumber ::= IA5String(SIZE(16))

**Value assignment:**

The approval number shall be provided as published on the corresponding European Commission web site, i.e. for example including hyphens if any. The approval number shall be left-aligned.

## 2.173. VuCalibrationData

Generation 1:

Information, stored in a vehicle unit, related to the calibrations of the recording equipment (Annex 1B requirement 098).

```
VuCalibrationData ::= SEQUENCE {
    noOfVuCalibrationRecords      INTEGER(0..255),
    vuCalibrationRecords          SET SIZE(noOfVuCalibrationRecords) OF VuCalibrationRecord
}
```

**noOfVuCalibrationRecords** is the number of records contained in the vuCalibrationRecords set.

**vuCalibrationRecords** is the set of calibration records.

## 2.174. VuCalibrationRecord

Information, stored in a vehicle unit, related a calibration of the recording equipment (Annex 1B requirement 098 and Annex 1C requirement 119 and 120).

Generation 1:

```
VuCalibrationRecord ::= SEQUENCE {
    calibrationPurpose             CalibrationPurpose,
    workshopName                   Name,
    workshopAddress                 Address,
    workshopCardNumber             FullCardNumber,
    workshopCardExpiryDate         TimeReal,
    vehicleIdentificationNumber     VehicleIdentificationNumber,
    vehicleRegistrationIdentification VehicleRegistrationIdentification,
    wVehicleCharacteristicConstant W-VehicleCharacteristicConstant,
    kConstantOfRecordingEquipment  K-ConstantOfRecordingEquipment,
    lTyreCircumference             L-TyreCircumference,
    tyreSize                       TyreSize,
    authorisedSpeed                 SpeedAuthorised,
    oldOdometerValue               OdometerShort,
    newOdometerValue               OdometerShort,
    oldTimeValue                   TimeReal,
    newTimeValue                   TimeReal,
    nextCalibrationDate            TimeReal
}
```

**calibrationPurpose** is the purpose of the calibration.

**workshopName, workshopAddress** are the workshop name and address.

**workshopCardNumber** identifies the workshop card used during the calibration.

**workshopCardExpiryDate** is the card expiry date.

**vehicleIdentificationNumber** is the VIN.

**vehicleRegistrationIdentification** contains the VRN and registering Member State.

**wVehicleCharacteristicConstant** is the characteristic coefficient of the vehicle.

**kConstantOfRecordingEquipment** is the constant of the recording equipment.

**ITyreCircumference** is the effective circumference of the wheel tyres.

**tyreSize** is the designation of the dimension of the tyres mounted on the vehicle

**authorisedSpeed** is the authorised speed of the vehicle.

**oldOdometerValue, newOdometerValue** are the old and new values of the odometer.

**oldTimeValue, newTimeValue** are the old and new values of date and time.

**nextCalibrationDate** is the date of the next calibration of the type specified in CalibrationPurpose to be carried out by the authorised inspection authority.

Generation 2:

```

VuCalibrationRecord ::= SEQUENCE {
    calibrationPurpose           CalibrationPurpose,
    workshopName                 Name,
    workshopAddress              Address,
    workshopCardNumber           FullCardNumber,
    workshopCardExpiryDate       TimeReal,
    vehicleIdentificationNumber   VehicleIdentificationNumber,
    vehicleRegistrationIdentification VehicleRegistrationIdentification,
    wVehicleCharacteristicConstant W-VehicleCharacteristicConstant,
    kConstantOfRecordingEquipment K-ConstantOfRecordingEquipment,
    lTyreCircumference           L-TyreCircumference,
    tyreSize                     TyreSize,
    authorisedSpeed               SpeedAuthorised,
    oldOdometerValue             OdometerShort,
    newOdometerValue             OdometerShort,
    oldTimeValue                 TimeReal,
    newTimeValue                 TimeReal,
    nextCalibrationDate          TimeReal,
    sealDataVu                   SealDataVu
}

```

In addition to generation 1 the following data element is used:

**sealDataVu** gives information about the seals that are attached to different components of the vehicle.

#### 2.175. VuCalibrationRecordArray

Generation 2:

Information, stored in a vehicle unit, related to the calibrations of the recording equipment (Annex 1C requirement 119 and 120).

```
VuCalibrationRecordArray ::= SEQUENCE {  
    recordType          RecordType,  
    recordSize          INTEGER(1..65535),  
    noOfRecords         INTEGER(0..65535),  
    records              SET SIZE(noOfRecords) OF VuCalibrationRecord  
}
```

**recordType** denotes the type of the record (VuCalibrationRecord). **Value Assignment:** See RecordType  
**recordSize** is the size of the VuCalibrationRecord in bytes.  
**noOfRecords** is the number of records in the set records.  
**records** is the set of calibration records.

#### 2.176. VuCardIWData

Generation 1:

Information, stored in a vehicle unit, related to insertion and withdrawal cycles of driver cards or of workshop cards in the vehicle unit (Annex 1B requirement 081 and Annex 1C requirement 103).

```
VuCardIWData ::= SEQUENCE {  
    noOfIWRecords      INTEGER(0..216-1),  
    vuCardIWRecords    SET SIZE(noOfIWRecords) OF VuCardIWRecord  
}
```

**noOfIWRecords** is the number of records in the set vuCardIWRecords.  
**vuCardIWRecords** is a set of records related to card insertion withdrawal cycles.



## 2.177. VuCardIWRecord

Information, stored in a vehicle unit, related to an insertion and withdrawal cycle of a driver card or of a workshop card in the vehicle unit (Annex 1B requirement 081 and Annex 1C requirement 102).

Generation 1:

```
VuCardIWRecord ::= SEQUENCE {
    cardHolderName          HolderName,
    fullCardNumber          FullCardNumber,
    cardExpiryDate          TimeReal,
    cardInsertionTime       TimeReal,
    vehicleOdometerValueAtInsertion OdometerShort,
    cardSlotNumber          CardSlotNumber,
    cardWithdrawalTime      TimeReal,
    vehicleOdometerValueAtWithdrawal OdometerShort,
    previousVehicleInfo      PreviousVehicleInfo,
    manualInputFlag         ManualInputFlag
}
```

**cardHolderName** is the driver or workshop card holder's surname and first names as stored in the card.

**fullCardNumber** is the type of card, its issuing Member State and its card number as stored in the card.

**cardExpiryDate** is the card's expiry date as stored in the card.

**cardInsertionTime** is the insertion date and time.

**vehicleOdometerValueAtInsertion** is the vehicle odometer value at card insertion.

**cardSlotNumber** is the slot in which the card is inserted.

**cardWithdrawalTime** is the withdrawal date and time.

**vehicleOdometerValueAtWithdrawal** is the vehicle odometer value at card withdrawal.

**previousVehicleInfo** contains information about the previous vehicle used by the driver, as stored in the card.

**manualInputFlag** is a flag identifying if the cardholder has manually entered driver activities at card insertion.

Generation 2:

```
VuCardIWRecord ::= SEQUENCE {  
    cardHolderName           HolderName,  
    fullCardNumberAndGeneration FullCardNumberAndGeneration,  
    cardExpiryDate           TimeReal,  
    cardInsertionTime        TimeReal,  
    vehicleOdometerValueAtInsertion OdometerShort,  
    cardSlotNumber           CardSlotNumber,  
    cardWithdrawalTime       TimeReal,  
    vehicleOdometerValueAtWithdrawal OdometerShort,  
    previousVehicleInfo      PreviousVehicleInfo,  
    manualInputFlag          ManualInputFlag  
}
```

Instead of fullCardNumber the generation 2 data structure makes use of the following data element.

**fullCardNumberAndGeneration** is the type of card, its issuing Member State, its card number and generation as stored in the card.

#### 2.178. VuCardIWRecordArray

Generation 2:

Information, stored in a vehicle unit, related to insertion and withdrawal cycles of driver cards or of workshop cards in the vehicle unit (Annex 1C requirement 103).

```
VuCardIWRecordArray ::= SEQUENCE {  
    recordType           RecordType,  
    recordSize           INTEGER(1..65535),  
    noOfRecords         INTEGER(0..65535),  
    records              SET SIZE(noOfRecords) OF VuCardIWRecord  
}
```

**recordType** denotes the type of the record (VuCardIWRecord). **Value Assignment:** See RecordType

**recordSize** is the size of the VuCardIWRecord in bytes.

**noOfRecords** is the number of records in the set records.

**records** is a set of records related to card insertion withdrawal cycles.

#### 2.179. VuCardRecord

Generation 2:

Information, stored in a vehicle unit, about a tachograph card used (Annex 1C requirement 132).

```

VuCardRecord ::= SEQUENCE {
    cardExtendedSerialNumber      ExtendedSerialNumber,
    cardPersonaliserID            OCTET STRING(SIZE(1)),
    typeOfTachographCardID       EquipmentType,
    cardStructureVersion          CardStructureVersion,
    cardNumber                    CardNumber
}

```

**cardExtendedSerialNumber** as read from the file EF\_ICC under the MF of the card.

**cardPersonaliserID** as read from the file EF\_ICC under the MF of the card.

**typeOfTachographCardId** as read from the file EF\_Application\_Identification under the DF\_Tachograph\_G2

**cardStructureVersion** as read from the file EF\_Application\_Identification under the DF\_Tachograph\_G2.

**cardNumber** as read from the file EF\_Identification under the DF\_Tachograph\_G2.

#### 2.180. VuCardRecordArray

Generation 2:

Information stored in a vehicle unit about the tachograph cards used with this VU. This information is intended for the analysis of VU – card problems (Annex 1C requirement 132).

```

VuCardRecordArray ::= SEQUENCE {
    recordType                    RecordType,
    recordSize                    INTEGER(1..65535),
    noOfRecords                  INTEGER(0..65535),
    records                      SET SIZE(noOfRecords) OF VuCardRecord
}

```

**recordType** denotes the type of the record (VuCardRecord). **Value Assignment:** See RecordType

**recordSize** is the size of the VuCardRecord in bytes.

**noOfRecords** is the number of records in the set records.

**records** is a set of records related to the tachograph cards used with the VU.

#### 2.181. VuCertificate

Certificate of the public key of a vehicle unit.

VuCertificate ::= Certificate

#### 2.182. VuCertificateRecordArray

Generation 2:

The VU certificate plus metadata as used in the download protocol.

```
VuCertificateRecordArray ::= SEQUENCE {  
    recordType          RecordType,  
    recordSize          INTEGER(1..65535),  
    noOfRecords         INTEGER(0..65535),  
    records              SET SIZE(noOfRecords) OF VuCertificate  
}
```

**recordType** denotes the type of the record (VuCertificate). **Value Assignment:** See RecordType

**recordSize** is the size of the VuCertificate in bytes.

**noOfRecords** is the number of records in the set records. The value shall be set to 1 as the certificates may have different lengths.

**records** is a set of VU certificates.

#### 2.183. VuCompanyLocksData

Generation 1:

Information, stored in a vehicle unit, related to company locks (Annex 1B requirement 104).

```
VuCompanyLocksData ::= SEQUENCE {  
    noOfLocks           INTEGER(0..255),  
    vuCompanyLocksRecords SET SIZE(noOfLocks) OF VuCompanyLocksRecord  
}
```

**noOfLocks** is the number of locks listed in vuCompanyLocksRecords.

**vuCompanyLocksRecords** is the set of company locks records.

## 2.184. VuCompanyLocksRecord

Information, stored in a vehicle unit, related to one company lock (Annex 1B requirement 104 and Annex 1C requirement 128).

Generation 1:

```
VuCompanyLocksRecord ::= SEQUENCE {
    lockInTime           TimeReal,
    lockOutTime          TimeReal,
    companyName          Name,
    companyAddress       Address,
    companyCardNumber    FullCardNumber
}
```

**lockInTime**, **lockOutTime** are the date and time of lock-in and lock-out.

**companyName**, **companyAddress** are the company name and address related with the lock-in.

**companyCardNumber** identifies the card used at lock-in.

Generation 2:

```
VuCompanyLocksRecord ::= SEQUENCE {
    lockInTime           TimeReal,
    lockOutTime          TimeReal,
    companyName          Name,
    companyAddress       Address,
    companyCardNumberAndGeneration FullCardNumberAndGeneration
}
```

Instead of **companyCardNumber** the generation 2 data structure makes use of the following data element.

**companyCardNumberAndGeneration** identifies the card including its generation used at lock-in.

## 2.185. VuCompanyLocksRecordArray

Generation 2:

Information, stored in a vehicle unit, related to company locks (Annex 1C requirement 128).

```
VuCompanyLocksRecordArray ::= SEQUENCE {  
    recordType          RecordType,  
    recordSize          INTEGER(1..65535),  
    noOfRecords         INTEGER(0..65535),  
    records             SET SIZE(noOfRecords) OF VuCompanyLocksRecord  
}
```

**recordType** denotes the type of the record (VuCompanyLocksRecord). **Value Assignment:** See RecordType

**recordSize** is the size of the VuCompanyLocksRecord in bytes.

**noOfRecords** is the number of records in the set records. Value 0..255.

**records** is the set of company locks records.

#### 2.186. VuControlActivityData

Generation 1:

Information, stored in a vehicle unit, related to controls performed using this VU (Annex 1B requirement 102).

```
VuControlActivityData ::= SEQUENCE {  
    noOfControls        INTEGER(0..20),  
    vuControlActivityRecords SET SIZE(noOfControls) OF VuControlActivityRecord  
}
```

**noOfControls** is the number of controls listed in vuControlActivityRecords.

**vuControlActivityRecords** is the set of control activity records.

#### 2.187. VuControlActivityRecord

Information, stored in a vehicle unit, related to a control performed using this VU (Annex 1B requirement 102 and Annex 1C requirement 126).

Generation 1:

```
VuControlActivityRecord ::= SEQUENCE {  
    controlType          ControlType,  
    controlTime          TimeReal,  
    controlCardNumber    FullCardNumber,  
    downloadPeriodBeginTime TimeReal,  
    downloadPeriodEndTime TimeReal  
}
```

**controlType** is the type of the control.

**controlTime** is the date and time of the control.

**controlCardNumber** identifies the control card used for the control.

**downloadPeriodBeginTime** is the begin time of the downloaded period, in case of downloading.

**downloadPeriodEndTime** is the end time of the downloaded period, in case of downloading.

Generation 2:

```
VuControlActivityRecord ::= SEQUENCE {
    controlType                ControlType,
    controlTime                TimeReal,
    controlCardNumberAndGeneration FullCardNumberAndGeneration,
    downloadPeriodBeginTime    TimeReal,
    downloadPeriodEndTime      TimeReal
}
```

Instead of controlCardNumber the generation 2 data structure makes use of the following data element.

**controlCardNumberAndGeneration** identifies the control card including its generation used for the control.

#### 2.188. VuControlActivityRecordArray

Generation 2:

Information, stored in a vehicle unit, related to controls performed using this VU (Annex 1C requirement 126).

```
VuControlActivityRecordArray ::= SEQUENCE {
    recordType                RecordType,
    recordSize                INTEGER(1..65535),
    noOfRecords              INTEGER(0..65535),
    records                   SET SIZE(noOfRecords) OF VuControlActivityRecord
}
```

**recordType** denotes the type of the record (VuControlActivityRecord). **Value Assignment:** See RecordType

**recordSize** is the size of the VuControlActivityRecord in bytes.

**noOfRecords** is the number of records in the set records.

**records** is the set of VU control activity records.

#### 2.189. VuDataBlockCounter

Counter, stored in a card, identifying sequentially the insertion withdrawal cycles of the card in vehicle units.

VuDataBlockCounter ::= BCDString(SIZE(2))

**Value assignment:** Consecutive Number with max, value 9 999, starting again with 0.

## 2.190. VuDetailedSpeedBlock

Information, stored in a vehicle unit, related to the vehicle's detailed speed for a minute during which the vehicle has been moving (Annex 1B requirement 093 and Annex 1C requirement 116).

```
VuDetailedSpeedBlock ::= SEQUENCE {  
    speedBlockBeginDate      TimeReal,  
    speedsPerSecond          SEQUENCE SIZE(60) OF Speed  
}
```

**speedBlockBeginDate** is the date and time of the first speed value within the block.

**speedsPerSecond** is the chronological sequence of measured speeds every seconds for the minute starting at speedBlockBeginDate (included).

## 2.191. VuDetailedSpeedBlockRecordArray

Generation 2:

Information, stored in a vehicle unit, related to the detailed speed of the vehicle.

```
VuDetailedSpeedBlockRecordArray ::= SEQUENCE {  
    recordType                RecordType,  
    recordSize                INTEGER(1..65535),  
    noOfRecords              INTEGER(0..65535),  
    records                   SET SIZE(noOfRecords) OF VuDetailedSpeedBlock  
}
```

**recordType** denotes the type of the record (VuDetailedSpeedBlock). **Value Assignment:** See RecordType

**recordSize** is the size of the VuDetailedSpeedBlock in bytes.

**noOfRecords** is the number of records in the set records.

**records** is the set of detailed speed blocks.

## 2.192. VuDetailedSpeedData

Generation 1:

Information, stored in a vehicle unit, related to the detailed speed of the vehicle.

```
VuDetailedSpeedData ::= SEQUENCE {  
    noOfSpeedBlocks          INTEGER(0..216-1),  
    vuDetailedSpeedBlocks    SET SIZE(noOfSpeedBlocks) OF VuDetailedSpeedBlock  
}
```

**noOfSpeedBlocks** is the number of speed blocks in the vuDetailedSpeedBlocks set.

**vuDetailedSpeedBlocks** is the set of detailed speed blocks.



## 2.193. VuDownloadablePeriod

Oldest and latest dates for which a vehicle unit holds data related to drivers activities (Annex 1B requirements 081, 084 or 087 and Annex 1C requirements 102, 105, 108).

```
VuDownloadablePeriod ::= SEQUENCE {
    minDownloadableTime      TimeReal
    maxDownloadableTime      TimeReal
}
```

**minDownloadableTime** is the oldest card insertion or activity change or place entry date and time stored in the VU.

**maxDownloadableTime** is the latest card withdrawal or activity change or place entry date and time stored in the VU.

## 2.194. VuDownloadablePeriodRecordArray

Generation 2:

The VuDownloadablePeriod plus metadata used in the download protocol.

```
VuDownloadablePeriodRecordArray ::= SEQUENCE {
    recordType                RecordType,
    recordSize                INTEGER(1..65535),
    noOfRecords               INTEGER(0..65535),
    records                   SET SIZE(noOfRecords) OF VuDownloadablePeriod
}
```

**recordType** denotes the type of the record (VuDownloadablePeriod). **Value Assignment:** See RecordType

**recordSize** is the size of the VuDownloadablePeriod in bytes.

**noOfRecords** is the number of records in the set records.

**records** is the set of VuDownloadablePeriod records.

## 2.195. VuDownloadActivityData

Information, stored in a vehicle unit, related to its last download (Annex 1B requirement 105 and Annex 1C requirement 129).

Generation 1:

```
VuDownloadActivityData ::= SEQUENCE {  
    downloadingTime          TimeReal,  
    fullCardNumber           FullCardNumber,  
    companyOrWorkshopName    Name  
}
```

**downloadingTime** is the date and time of downloading.

**fullCardNumber** identifies the card used to authorise the download.

**companyOrWorkshopName** is the company or workshop name.

Generation 2:

```
VuDownloadActivityData ::= SEQUENCE {  
    downloadingTime          TimeReal,  
    fullCardNumberAndGeneration FullCardNumberAndGeneration,  
    companyOrWorkshopName    Name  
}
```

Instead of fullCardNumber the generation 2 data structure makes use of the following data element.

**fullCardNumberAndGeneration** identifies the card including its generation used to authorise the download.

## 2.196. VuDownloadActivityDataRecordArray

Generation 2:

Information related to the last VU download (Annex 1C requirement 129).

```
VuDownloadActivityDataRecordArray ::= SEQUENCE {  
    recordType                RecordType,  
    recordSize                 INTEGER(1..65535),  
    noOfRecords                INTEGER(0..65535),  
    records                    SET SIZE(noOfRecords) OF VuDownloadActivityData  
}
```

**recordType** denotes the type of the record (VuDownloadActivityData). **Value Assignment:** See RecordType

**recordSize** is the size of the VuDownloadActivityData in bytes.

**noOfRecords** is the number of records in the set records.

**records** is the set of download activity data records.

## 2.197. VuEventData

Generation 1:

Information, stored in a vehicle unit, related to events (Annex 1B requirement 094 except over speeding event).

```
VuEventData ::= SEQUENCE {
    noOfVuEvents          INTEGER(0..255),
    vuEventRecords        SET SIZE(noOfVuEvents) OF VuEventRecord
}
```

**noOfVuEvents** is the number of events listed in the vuEventRecords set.

**vuEventRecords** is a set of events records.

## 2.198. VuEventRecord

Information, stored in a vehicle unit, related to an event (Annex 1B requirement 094 and Annex 1C requirement 117 except over speeding event).

Generation 1:

```
VuEventRecord ::= SEQUENCE {
    eventType              EventFaultType,
    eventRecordPurpose     EventFaultRecordPurpose,
    eventBeginTime         TimeReal,
    eventEndTime           TimeReal,
    cardNumberDriverSlotBegin FullCardNumber,
    cardNumberCodriverSlotBegin FullCardNumber,
    cardNumberDriverSlotEnd FullCardNumber,
    cardNumberCodriverSlotEnd FullCardNumber,
    similarEventsNumber    SimilarEventsNumber
}
```

**eventType** is the type of the event.

**eventRecordPurpose** is the purpose for which this event has been recorded.

**eventBeginTime** is the date and time of beginning of event.

**eventEndTime** is the date and time of end of event.

**cardNumberDriverSlotBegin** identifies the card inserted in the driver slot at the beginning of the event.

**cardNumberCodriverSlotBegin** identifies the card inserted in the co-driver slot at the beginning of the event.

**cardNumberDriverSlotEnd** identifies the card inserted in the driver slot at the end of the event.

**cardNumberCodriverSlotEnd** identifies the card inserted in the co-driver slot at the end of the event.

**similarEventsNumber** is the number of similar events that day.

This sequence can be used for all events other than over speeding events.

Generation 2:

```
VuEventRecord ::= SEQUENCE {  
    eventType                EventFaultType,  
    eventRecordPurpose       EventFaultRecordPurpose,  
    eventBeginTime           TimeReal,  
    eventEndTime             TimeReal,  
    cardNumberAndGenDriverSlotBegin FullCardNumberAndGeneration,  
    cardNumberAndGenCodriverSlotBegin FullCardNumberAndGeneration,  
    cardNumberAndGenDriverSlotEnd FullCardNumberAndGeneration,  
    cardNumberAndGenCodriverSlotEnd FullCardNumberAndGeneration,  
    similarEventsNumber      SimilarEventsNumber,  
    manufacturerSpecificEventFaultData ManufacturerSpecificEventFaultData  
}
```

In addition to generation 1 the following data elements are used:

**manufacturerSpecificEventFaultData** contains additional, manufacturer specific information about the event.

Instead of **cardNumberDriverSlotBegin**, **cardNumberCodriverSlotBegin**, **cardNumberDriverSlotEnd**, and **cardNumberCodriverSlotEnd** the generation 2 data structure makes use of the following data elements:

**cardNumberAndGenDriverSlotBegin** identifies the card including its generation which is inserted in the driver slot at the beginning of the event.

**cardNumberAndGenCodriverSlotBegin** identifies the card including its generation which is inserted in the co-driver slot at the beginning of the event.

**cardNumberAndGenDriverSlotEnd** identifies the card including its generation which is inserted in the driver slot at the end of the event.

**cardNumberAndGenCodriverSlotEnd** identifies the card including its generation which is inserted in the co-driver slot at the end of the event.

If the event is a time conflict the **eventBeginTime** and **eventEndTime** are to be interpreted as follows:

**eventBeginTime** is the recording equipment date and time.

**eventEndTime** is the GNSS date and time.

#### 2.199. VuEventRecordArray

Generation 2:

Information, stored in a vehicle unit, related to events (Annex 1C requirement 117 except over speeding event).

```

VuEventRecordArray ::= SEQUENCE {
    recordType          RecordType,
    recordSize          INTEGER(1..65535),
    noOfRecords        INTEGER(0..65535),
    records             SET SIZE(noOfRecords) OF VuEventRecord
}

```

**recordType** denotes the type of the record (VuEventRecord). **Value Assignment:** See RecordType

**recordSize** is the size of the VuEventRecord in bytes.

**noOfRecords** is the number of records in the set records.

**records** is a set of events records.

#### 2.200. VuFaultData

Generation 1:

Information, stored in a vehicle unit, related to faults (Annex 1B requirement 096).

```

VuFaultData ::= SEQUENCE {
    noOfVuFaults       INTEGER(0..255),
    vuFaultRecords     SET SIZE(noOfVuFaults) OF VuFaultRecord
}

```

**noOfVuFaults** is the number of faults listed in the vuFaultRecords set.

**vuFaultRecords** is a set of faults records.

## 2.201. VuFaultRecord

Information, stored in a vehicle unit, related to a fault (Annex 1B requirement 096 and Annex 1C requirement 118).

Generation 1:

```
VuFaultRecord ::= SEQUENCE {  
    faultType                EventFaultType,  
    faultRecordPurpose       EventFaultRecordPurpose,  
    faultBeginTime           TimeReal,  
    faultEndTime             TimeReal,  
    cardNumberDriverSlotBegin FullCardNumber,  
    cardNumberCodriverSlotBegin FullCardNumber,  
    cardNumberDriverSlotEnd FullCardNumber,  
    cardNumberCodriverSlotEnd FullCardNumber  
}
```

**faultType** is the type of recording equipment fault.

**faultRecordPurpose** is the purpose for which this fault has been recorded.

**faultBeginTime** is the date and time of beginning of fault.

**faultEndTime** is the date and time of end of fault.

**cardNumberDriverSlotBegin** identifies the card inserted in the driver slot at the beginning of the fault.

**cardNumberCodriverSlotBegin** identifies the card inserted in the co-driver slot at the beginning of the fault.

**cardNumberDriverSlotEnd** identifies the card inserted in the driver slot at the end of the fault.

**cardNumberCodriverSlotEnd** identifies the card inserted in the co-driver slot at the end of the fault.

Generation 2:

```
VuFaultRecord ::= SEQUENCE {  
    faultType                EventFaultType,  
    faultRecordPurpose       EventFaultRecordPurpose,  
    faultBeginTime           TimeReal,  
    faultEndTime             TimeReal,  
    cardNumberAndGenDriverSlotBegin FullCardNumberAndGeneration,  
    cardNumberAndGenCodriverSlotBegin FullCardNumberAndGeneration,  
    cardNumberAndGenDriverSlotEnd FullCardNumberAndGeneration,  
    cardNumberAndGenCodriverSlotEnd FullCardNumberAndGeneration,  
    manufacturerSpecificEventFaultData ManufacturerSpecificEventFaultData  
}
```

In addition to generation 1 the following data element is used:

**manufacturerSpecificEventFaultData** contains additional, manufacturer specific information about the fault.

Instead of `cardNumberDriverSlotBegin`, `cardNumberCodriverSlotBegin`, `cardNumberDriverSlotEnd`, and `cardNumberCodriverSlotEnd` the generation 2 data structure makes use of the following data elements:

**cardNumberAndGenDriverSlotBegin** identifies the card including its generation which is inserted in the driver slot at the beginning of the fault.

**cardNumberAndGenCodriverSlotBegin** identifies the card including its generation which is inserted in the co-driver slot at the beginning of the fault.

**cardNumberAndGenDriverSlotEnd** identifies the card including its generation which is inserted in the driver slot at the end of the fault.

**cardNumberAndGenCodriverSlotEnd** identifies the card including its generation which is inserted in the co-driver slot at the end of the fault.

#### 2.202. VuFaultRecordArray

Generation 2:

Information, stored in a vehicle unit, related to faults (Annex 1C requirement 118).

VuFaultRecordArray ::= SEQUENCE {

<code>recordType</code>	<code>RecordType</code> ,
<code>recordSize</code>	<code>INTEGER(1..65535)</code> ,
<code>noOfRecords</code>	<code>INTEGER(0..65535)</code> ,
<code>records</code>	<code>SET SIZE(noOfRecords) OF VuFaultRecord</code>

}

**recordType** denotes the type of the record (VuFaultRecord). **Value Assignment:** See RecordType

**recordSize** is the size of the VuFaultRecord in bytes.

**noOfRecords** is the number of records in the set records.

**records** is a set of faults records.

#### 2.203. VuGNSSCDRecord

Generation 2:

Information, stored in a vehicle unit, related to the GNSS position of the vehicle if the continuous driving time of the driver reaches a multiple of three hours (Annex 1C requirement 108, 110).

VuGNSSCDRecord ::= SEQUENCE {

timeStamp	TimeReal,
cardNumberAndGenDriverSlot	FullCardNumberAndGeneration,
cardNumberAndGenCodriverSlot	FullCardNumberAndGeneration,
gnssPlaceRecord	GNSSPlaceRecord

}

**timeStamp** is the date and time when the continuous driving time of the card holder reaches a multiple of three hours.

**cardNumberAndGenDriverSlot** identifies the card including its generation which is inserted in the driver slot.

**cardNumberAndGenCodriverSlot** identifies the card including its generation which is inserted in the co-driver slot.

**gnssPlaceRecord** contains information related to the position of the vehicle.

#### 2.204. VuGNSSCDRecordArray

Generation 2:

Information, stored in a vehicle unit, related to the GNSS position of the vehicle if the continuous driving time of the driver reaches a multiple of three hours (Annex 1C requirement 108 and 110).

VuGNSSCDRecordArray ::= SEQUENCE {

recordType	RecordType,
recordSize	INTEGER(1..65535),
noOfRecords	INTEGER(0..65535),
records	SET SIZE(noOfRecords) OF VuGNSSCDRecord

}

**recordType** denotes the type of the record (VuGNSSCDRecord). **Value Assignment:** See RecordType

**recordSize** is the size of the VuGNSSCDRecord in bytes.

**noOfRecords** is the number of records in the set records.

**records** is a set of GNSS continuous driving records.



## 2.205. VuIdentification

Information, stored in a vehicle unit, related to the identification of the vehicle unit (Annex 1B requirement 075 and Annex 1C requirement 93 and 121).

Generation 1:

```
VuIdentification ::= SEQUENCE {
    vuManufacturerName          VuManufacturerName,
    vuManufacturerAddress       VuManufacturerAddress,
    vuPartNumber                VuPartNumber,
    vuSerialNumber              VuSerialNumber,
    vuSoftwareIdentification    VuSoftwareIdentification,
    vuManufacturingDate         VuManufacturingDate,
    vuApprovalNumber            VuApprovalNumber
}
```

**vuManufacturerName** is the name of the manufacturer of the vehicle unit.

**vuManufacturerAddress** is the address of the manufacturer of the vehicle unit.

**vuPartNumber** is the part number of the vehicle unit.

**vuSerialNumber** is the serial number of the vehicle unit.

**vuSoftwareIdentification** identifies the software implemented in the vehicle unit.

**vuManufacturingDate** is the manufacturing date of the vehicle unit.

**vuApprovalNumber** is the type approval number of the vehicle unit.

Generation 2:

```
VuIdentification ::= SEQUENCE {
    vuManufacturerName          VuManufacturerName,
    vuManufacturerAddress       VuManufacturerAddress,
    vuPartNumber                VuPartNumber,
    vuSerialNumber              VuSerialNumber,
    vuSoftwareIdentification    VuSoftwareIdentification,
    vuManufacturingDate         VuManufacturingDate,
    vuApprovalNumber            VuApprovalNumber,
    vuGeneration                Generation,
    vuAbility                    VuAbility
}
```

In addition to generation 1 the following data element are used:

**vuGeneration** identifies the generation of the vehicle unit.

**vuAbility** provides information whether the VU supports generation 1 tachograph cards or not.

#### 2.206. VuIdentificationRecordArray

Generation 2:

The VuIdentification plus metadata used in the download protocol.

VuIdentificationRecordArray ::= SEQUENCE {

recordType	RecordType,
recordSize	INTEGER(1..65535),
noOfRecords	INTEGER(0..65535),
records	SET SIZE(noOfRecords) OF VuIdentification

}

**recordType** denotes the type of the record (VuIdentification). **Value Assignment:** See RecordType

**recordSize** is the size of the VuIdentification in bytes.

**noOfRecords** is the number of records in the set records.

**records** is a set of VuIdentification records.

#### 2.207. VuITSConsentRecord

Generation 2:

Information stored in a vehicle unit, related to the consent of a driver to use Intelligent Transport Systems.

VuITSConsentRecord ::= SEQUENCE {

cardNumberAndGen	FullCardNumberAndGeneration,
consent	BOOLEAN

}

**cardNumberAndGen** identifies the card including its generation. This must be a driver card or a workshop card.

**consent** is a flag which indicates whether the driver has given his consent on the usage of Intelligent Transport Systems with this vehicle / vehicle unit.

**Value assignment:**

TRUE indicates the driver's consent to use Intelligent Transport Systems

FALSE indicates the driver's denial to use Intelligent Transport Systems

#### 2.208. VuITSConsentRecordArray

Generation 2:

Information, stored in a vehicle unit, related to drivers' consent on the usage of Intelligent Transport Systems (Annex 1C requirement 200).

```

VuITSConsentRecordArray ::= SEQUENCE {
    recordType          RecordType,
    recordSize          INTEGER(1..65535),
    noOfRecords         INTEGER(0..65535),
    records              SET SIZE(noOfRecords) OF VuITSConsentRecord
}

```

**recordType** denotes the type of the record (VuITSConsentRecord). **Value Assignment:** See RecordType

**recordSize** is the size of the VuITSConsentRecord in bytes.

**noOfRecords** is the number of records in the set records.

**records** is the set of ITS consent records.

#### 2.209. VuManufacturerAddress

Address of the manufacturer of the vehicle unit.

VuManufacturerAddress ::= Address

**Value assignment:** Unspecified.

#### 2.210. VuManufacturerName

Name of the manufacturer of the vehicle unit.

VuManufacturerName ::= Name

**Value assignment:** Unspecified.

#### 2.211. VuManufacturingDate

Date of manufacture of the vehicle unit.

VuManufacturingDate ::= TimeReal

**Value assignment:** Unspecified.

#### 2.212. VuOverSpeedingControlData

Information, stored in a vehicle unit, related to over speeding events since the last over speeding control (Annex 1B requirement 095 and Annex 1C requirement 117).

```

VuOverSpeedingControlData ::= SEQUENCE {
    lastOverspeedControlTime    TimeReal,
    firstOverspeedSince         TimeReal,
    numberOfOverspeedSince      OverspeedNumber
}

```

**lastOverspeedControlTime** is the date and time of the last over speeding control.

**firstOverspeedSince** is the date and time of the first over speeding following this over speeding control.

**numberOfOverspeedSince** is the number of over speeding events since the last over speeding control.

## 2.213. VuOverSpeedingControlDataRecordArray

Generation 2:

The VuOverSpeedingControlData plus metadata used in the download protocol.

```
VuOverSpeedingControlDataRecordArray ::= SEQUENCE {  
    recordType          RecordType,  
    recordSize          INTEGER(1..65535),  
    noOfRecords         INTEGER(0..65535),  
    records              SET SIZE(noOfRecords) OF VuOverSpeedingControlData  
}
```

**recordType** denotes the type of the record (VuOverSpeedingControlData). **Value Assignment:** See RecordType

**recordSize** is the size of the VuOverSpeedingControlData in bytes.

**noOfRecords** is the number of records in the set records.

**records** is a set of over speeding control data records.

## 2.214. VuOverSpeedingEventData

Generation 1:

Information, stored in a vehicle unit, related to over speeding events (Annex 1B requirement 094).

```
VuOverSpeedingEventData ::= SEQUENCE {  
    noOfVuOverSpeedingEvents    INTEGER(0..255),  
    vuOverSpeedingEventRecords  SET SIZE(noOfVuOverSpeedingEvents) OF  
                                VuOverSpeedingEventRecord  
}
```

**noOfVuOverSpeedingEvents** is the number of events listed in the vuOverSpeedingEventRecords set.

**vuOverSpeedingEventRecords** is a set of over speeding events records.

## 2.215. VuOverSpeedingEventRecord

Generation 1:

Information, stored in a vehicle unit, related to over speeding events (Annex 1B requirement 094 and Annex 1C requirement 117).

```
VuOverSpeedingEventRecord ::= SEQUENCE {
    eventType                EventFaultType,
    eventRecordPurpose       EventFaultRecordPurpose,
    eventBeginTime           TimeReal,
    eventEndTime             TimeReal,
    maxSpeedValue            SpeedMax,
    averageSpeedValue        SpeedAverage,
    cardNumberDriverSlotBegin FullCardNumber,
    similarEventsNumber      SimilarEventsNumber
}
```

**eventType** is the type of the event.

**eventRecordPurpose** is the purpose for which this event has been recorded.

**eventBeginTime** is the date and time of beginning of event.

**eventEndTime** is the date and time of end of event.

**maxSpeedValue** is the maximum speed measured during the event.

**averageSpeedValue** is the arithmetic average speed measured during the event.

**cardNumberDriverSlotBegin** identifies the card inserted in the driver slot at the beginning of the event.

**similarEventsNumber** is the number of similar events that day.

Generation 2:

Information, stored in a vehicle unit, related to over speeding events (Annex 1B requirement 094 and Annex 1C requirement 117).

```
VuOverSpeedingEventRecord ::= SEQUENCE {  
    eventType                EventFaultType,  
    eventRecordPurpose       EventFaultRecordPurpose,  
    eventBeginTime           TimeReal,  
    eventEndTime             TimeReal,  
    maxSpeedValue            SpeedMax,  
    averageSpeedValue        SpeedAverage,  
    cardNumberAndGenDriverSlotBegin FullCardNumberAndGeneration,  
    similarEventsNumber      SimilarEventsNumber  
}
```

Instead of `cardNumberDriverSlotBegin`, the generation 2 data structure makes use of the following data element:

**cardNumberAndGenDriverSlotBegin** identifies the card including its generation which is inserted in the driver slot at the beginning of the event.

#### 2.216. VuOverSpeedingEventRecordArray

Generation 2:

Information, stored in a vehicle unit, related to over speeding events (Annex 1C requirement 117).

```
VuOverSpeedingEventRecordArray ::= SEQUENCE {  
    recordType                RecordType,  
    recordSize                 INTEGER(1..65535),  
    noOfRecords               INTEGER(0..65535),  
    records                    SET SIZE(noOfRecords) OF VuOverSpeedingEventRecord  
}
```

**recordType** denotes the type of the record (`VuOverSpeedingEventRecord`). **Value Assignment:** See `RecordType`

**recordSize** is the size of the `VuOverSpeedingEventRecord` in bytes.

**noOfRecords** is the number of records in the set `records`.

**records** is a set of over speeding events records.

#### 2.217. VuPartNumber

Part number of the vehicle unit.

`VuPartNumber ::= IA5String(SIZE(16))`

**Value assignment:** VU manufacturer specific.

## 2.218. VuPlaceDailyWorkPeriodData

Generation 1:

Information, stored in a vehicle unit, related to places where drivers begin or end a daily work period (Annex 1B requirement 087 and Annex 1C requirement 108 and 110).

```
VuPlaceDailyWorkPeriodData ::= SEQUENCE {
    noOfPlaceRecords          INTEGER(0..255),
    vuPlaceDailyWorkPeriodRecords  SET SIZE(noOfPlaceRecords) OF
                                   VuPlaceDailyWorkPeriodRecord
}
```

**noOfPlaceRecords** is the number of records listed in the vuPlaceDailyWorkPeriodRecords set.

**vuPlaceDailyWorkPeriodRecords** is a set of place related records.

## 2.219. VuPlaceDailyWorkPeriodRecord

Generation 1:

Information, stored in a vehicle unit, related to a place where a driver begins or ends a daily work period (Annex 1B requirement 087 and Annex 1C requirement 108 and 110).

```
VuPlaceDailyWorkPeriodRecord ::= SEQUENCE {
    fullCardNumber            FullCardNumber,
    placeRecord                PlaceRecord
}
```

**fullCardNumber** is the driver's card type, card issuing Member State and card number.

**placeRecord** contains the information related to the place entered.

Generation 2:

Information, stored in a vehicle unit, related to a place where a driver begins or ends a daily work period (Annex 1B requirement 087 and Annex 1C requirement 108 and 110).

```
VuPlaceDailyWorkPeriodRecord ::= SEQUENCE {
    fullCardNumberAndGeneration  FullCardNumberAndGeneration,
    placeRecord                PlaceRecord
}
```

Instead of fullCardNumber, the generation 2 data structure makes use of the following data element:

**fullCardNumberAndGeneration** is the type of card, its issuing Member State, its card number and generation as stored in the card.

## 2.220. VuPlaceDailyWorkPeriodRecordArray

Generation 2:

Information, stored in a vehicle unit, related to places where drivers begin or end a daily work period (Annex 1C requirement 108 and 110).

```
VuPlaceDailyWorkPeriodRecordArray ::= SEQUENCE {  
    recordType          RecordType,  
    recordSize          INTEGER(1..65535),  
    noOfRecords         INTEGER(0..65535),  
    records             SET SIZE(noOfRecords) OF VuPlaceDailyWorkPeriodRecord  
}
```

**recordType** denotes the type of the record (VuPlaceDailyWorkPeriodRecord). **Value Assignment:** See RecordType

**recordSize** is the size of the VuPlaceDailyWorkPeriodRecord in bytes.

**noOfRecords** is the number of records in the set records.

**records** is a set of place related records.

#### 2.221. VuPrivateKey

Generation 1:

The private key of a vehicle unit.

VuPrivateKey ::= RSAKeyPrivateExponent

#### 2.222. VuPublicKey

Generation 1:

The public key of a vehicle unit.

VuPublicKey ::= PublicKey

#### 2.223. VuSerialNumber

Serial number of the vehicle unit (Annex 1B requirement 075 and Annex 1C requirement 93).

VuSerialNumber ::= ExtendedSerialNumber

#### 2.224. VuSoftInstallationDate

Date of installation of the vehicle unit software version.

VuSoftInstallationDate ::= TimeReal

**Value assignment:** Unspecified.



## 2.225. VuSoftwareIdentification

Information, stored in a vehicle unit, related to the software installed.

```
VuSoftwareIdentification ::= SEQUENCE {
    vuSoftwareVersion          VuSoftwareVersion,
    vuSoftInstallationDate     VuSoftInstallationDate
}
```

**vuSoftwareVersion** is the software version number of the Vehicle Unit.

**vuSoftInstallationDate** is the software version installation date.

## 2.226. VuSoftwareVersion

Software version number of the vehicle unit.

```
VuSoftwareVersion ::= IA5String(SIZE(4))
```

**Value assignment:** Unspecified.

## 2.227. VuSpecificConditionData

Generation 1:

Information, stored in a vehicle unit, related to specific conditions.

```
VuSpecificConditionData ::= SEQUENCE {
    noOfSpecificConditionRecords    INTEGER(0..216-1)
    specificConditionRecords        SET SIZE (noOfSpecificConditionRecords) OF
                                    SpecificConditionRecord
}
```

**noOfSpecificConditionRecords** is the number of records listed in the specificConditionRecords set.

**specificConditionRecords** is a set of specific conditions related records.

## 2.228. VuSpecificConditionRecordArray

Generation 2:

Information, stored in a vehicle unit, related to specific conditions (Annex 1C requirement 130).

```
VuSpecificConditionRecordArray ::= SEQUENCE {  
    recordType          RecordType,  
    recordSize          INTEGER(1..65535),  
    noOfRecords         INTEGER(0..65535),  
    records              SET SIZE(noOfRecords) OF SpecificConditionRecord  
}
```

**recordType** denotes the type of the record (SpecificConditionRecord). **Value Assignment:** See RecordType

**recordSize** is the size of the SpecificConditionRecord in bytes.

**noOfRecords** is the number of records in the set records.

**records** is a set of specific conditions related records.

#### 2.229. VuTimeAdjustmentData

Generation 1:

Information, stored in a vehicle unit, related to time adjustments performed outside the frame of a regular calibration (Annex 1B requirement 101).

```
VuTimeAdjustmentData ::= SEQUENCE {  
    noOfVuTimeAdjRecords    INTEGER(0..6),  
    vuTimeAdjustmentRecords SET SIZE(noOfVuTimeAdjRecords) OF  
                             VuTimeAdjustmentRecord  
}
```

**noOfVuTimeAdjRecords** is the number of records in vuTimeAdjustmentRecords.

**vuTimeAdjustmentRecords** is a set of time adjustment records.

#### 2.230. VuTimeAdjustmentGNSSRecord

Generation 2:

Information, stored in a vehicle unit, related to a time adjustment based on time data from GNSS (Annex 1C requirement 124 and 125).

```
VuTimeAdjustmentGNSSRecord ::= SEQUENCE {  
    oldTimeValue            TimeReal,  
    newTimeValue            TimeReal  
}
```

**oldTimeValue**, **newTimeValue** are the old and new values of date and time.

#### 2.231. VuTimeAdjustmentGNSSRecordArray

Generation 2:

Information, stored in a vehicle unit, related to a time adjustment performed bybased on time data from GNSS (Annex 1C requirement 124 and 125).

```
VuTimeAdjustmentGNSSRecordArray ::= SEQUENCE {
    recordType          RecordType,
    recordSize          INTEGER(1..65535),
    noOfRecords         INTEGER(0..65535),
    records             SET SIZE(noOfRecords) OF VuTimeAdjustmentGNSSRecord
}
```

**recordType** denotes the type of the record (VuTimeAdjustmentGNSSRecord). **Value Assignment:** See RecordType

**recordSize** is the size of the VuTimeAdjustmentGNSSRecord in bytes.

**noOfRecords** is the number of records in the set records.

**records** is a set of GNSS time adjustment records.

#### 2.232. VuTimeAdjustmentRecord

Information, stored in a vehicle unit, related a time adjustment performed outside the frame of a regular calibration (Annex 1B requirement 101 and Annex 1C requirement 124 and 125).

Generation 1:

```
VuTimeAdjustmentRecord ::= SEQUENCE {
    oldTimeValue        TimeReal,
    newTimeValue        TimeReal,
    workshopName        Name,
    workshopAddress     Address,
    workshopCardNumber FullCardNumber
}
```

**oldTimeValue**, **newTimeValue** are the old and new values of date and time.

**workshopName**, **workshopAddress** are the workshop name and address.

**workshopCardNumber** identifies the workshop card used to perform the time adjustment.

Generation 2:

```
VuTimeAdjustmentRecord ::= SEQUENCE {  
    oldTimeValue           TimeReal,  
    newTimeValue           TimeReal,  
    workshopName           Name,  
    workshopAddress        Address,  
    workshopCardNumberAndGeneration FullCardNumberAndGeneration  
}
```

Instead of workshopCardNumber the generation 2 data structure makes use of the following data element.

**workshopCardNumberAndGeneration** identifies the workshop card including its generation used to perform the time adjustment.

### 2.233. VuTimeAdjustmentRecordArray

Generation 2:

Information, stored in a vehicle unit, related to time adjustments performed outside the frame of a regular calibration (Annex 1C requirement 124 and 125).

```
VuTimeAdjustmentRecordArray ::= SEQUENCE {  
    recordType             RecordType,  
    recordSize             INTEGER(1..65535),  
    noOfRecords           INTEGER(0..65535),  
    records                SET SIZE(noOfRecords) OF VuTimeAdjustmentRecord  
}
```

**recordType** denotes the type of the record (VuTimeAdjustmentRecord). **Value Assignment:** See RecordType

**recordSize** is the size of the VuTimeAdjustmentRecord in bytes.

**noOfRecords** is the number of records in the set records.

**records** is a set of time adjustment records.

## 2.234. WorkshopCardApplicationIdentification

Information, stored in a workshop card related to the identification of the application of the card (Annex 1C requirement 307 and 330).

Generation 1:

```
WorkshopCardApplicationIdentification ::= SEQUENCE {
    typeOfTachographCardId           EquipmentType,
    cardStructureVersion              CardStructureVersion,
    noOfEventsPerType                 NoOfEventsPerType,
    noOfFaultsPerType                 NoOfFaultsPerType,
    activityStructureLength            CardActivityLengthRange,
    noOfCardVehicleRecords            NoOfCardVehicleRecords,
    noOfCardPlaceRecords              NoOfCardPlaceRecords,
    noOfCalibrationRecords            NoOfCalibrationRecords
}
```

**typeOfTachographCardId** is specifying the implemented type of card.

**cardStructureVersion** is specifying the the version of the structure that is implemented in the card.

**noOfEventsPerType** is the number of events per type of event the card can record.

**noOfFaultsPerType** is the number of faults per type of fault the card can record.

**activityStructureLength** indicates the number of bytes available for storing activity records.

**noOfCardVehicleRecords** is the number of vehicle records the card can contain.

**noOfCardPlaceRecords** is the number of places the card can record.

**noOfCalibrationRecords** is the number of calibration records the card can store.

Generation 2:

```
WorkshopCardApplicationIdentification ::= SEQUENCE {  
    typeOfTachographCardId           EquipmentType,  
    cardStructureVersion               CardStructureVersion,  
    noOfEventsPerType                 NoOfEventsPerType,  
    noOfFaultsPerType                 NoOfFaultsPerType,  
    activityStructureLength            CardActivityLengthRange,  
    noOfCardVehicleRecords            NoOfCardVehicleRecords,  
    noOfCardPlaceRecords              NoOfCardPlaceRecords,  
    noOfCalibrationRecords             NoOfCalibrationRecords,  
    noOfGNSSCDRecords                 NoOfGNSSCDRecords,  
    noOfSpecificConditionRecords       NoOfSpecificConditionRecords  
}
```

In addition to generation 1 the following data elements are used:

**noOfGNSSCDRecords** is the number of GNSS continuous driving records the card can store.

**noOfSpecificConditionRecords** is the number of specific condition records the card can store.

#### 2.235. WorkshopCardCalibrationData

Information, stored in a workshop card, related to workshop activity performed with the card (Annex 1C requirements 314, 316, 337, and 339).

```
WorkshopCardCalibrationData ::= SEQUENCE {  
    calibrationTotalNumber             INTEGER(0 .. 216-1),  
    calibrationPointerNewestRecord      INTEGER(0 .. NoOfCalibrationRecords-1),  
    calibrationRecords                  SET SIZE(NoOfCalibrationRecords) OF  
                                        WorkshopCardCalibrationRecord  
}
```

**calibrationTotalNumber** is the total number of calibrations performed with the card.

**calibrationPointerNewestRecord** is the index of the last updated calibration record.

**Value assignment:** Number corresponding to the numerator of the calibration record, beginning with '0' for the first occurrence of the calibration records in the structure.

**calibrationRecords** is the set of records containing calibration and/or time adjustment information.

## 2.236. WorkshopCardCalibrationRecord

Information, stored in a workshop card, related to a calibration performed with the card (Annex 1C requirement 314 and 337).

Generation 1:

```
WorkshopCardCalibrationRecord ::= SEQUENCE {
    calibrationPurpose           CalibrationPurpose,
    vehicleIdentificationNumber  VehicleIdentificationNumber,
    vehicleRegistration          VehicleRegistrationIdentification,
    wVehicleCharacteristicConstant W-VehicleCharacteristicConstant,
    kConstantOfRecordingEquipment K-ConstantOfRecordingEquipment,
    lTyreCircumference          L-TyreCircumference,
    tyreSize                    TyreSize,
    authorisedSpeed              SpeedAuthorised,
    oldOdometerValue            OdometerShort,
    newOdometerValue            OdometerShort,
    oldTimeValue                TimeReal,
    newTimeValue                TimeReal,
    nextCalibrationDate         TimeReal,
    vuPartNumber                VuPartNumber,
    vuSerialNumber              VuSerialNumber,
    sensorSerialNumber          SensorSerialNumber
}
```

**calibrationPurpose** is the purpose of the calibration.

**vehicleIdentificationNumber** is the VIN.

**vehicleRegistration** contains the VRN and registering Member State.

**wVehicleCharacteristicConstant** is the characteristic coefficient of the vehicle.

**kConstantOfRecordingEquipment** is the constant of the recording equipment.

**lTyreCircumference** is the effective circumference of the wheel tyres.

**tyreSize** is the designation of the dimensions of the tyres mounted on the vehicle.

**authorisedSpeed** is the maximum authorised speed of the vehicle.

**oldOdometerValue**, **newOdometerValue** are the old and new values of the odometer.

**oldTimeValue**, **newTimeValue** are the old and new values of date and time.

**nextCalibrationDate** is the date of the next calibration of the type specified in CalibrationPurpose to be carried out by the authorised inspection authority.

**vuPartNumber**, **vuSerialNumber** and **sensorSerialNumber** are the data elements for recording equipment identification.

Generation 2:

```
WorkshopCardCalibrationRecord ::= SEQUENCE {
    calibrationPurpose           CalibrationPurpose,
    vehicleIdentificationNumber  VehicleIdentificationNumber,
    vehicleRegistration          VehicleRegistrationIdentification,
    wVehicleCharacteristicConstant W-VehicleCharacteristicConstant,
    kConstantOfRecordingEquipment K-ConstantOfRecordingEquipment,
    lTyreCircumference          L-TyreCircumference,
    tyreSize                    TyreSize,
    authorisedSpeed              SpeedAuthorised,
    oldOdometerValue            OdometerShort,
    newOdometerValue            OdometerShort,
    oldTimeValue                TimeReal,
    newTimeValue                TimeReal,
    nextCalibrationDate         TimeReal,
    vuPartNumber                VuPartNumber,
    vuSerialNumber              VuSerialNumber,
    sensorSerialNumber          SensorSerialNumber,
    sensorGNSSSerialNumber      SensorGNSSSerialNumber,
    rcmSerialNumber             RemoteCommunicationModuleSerialNumber,
    sealDataCard                SealDataCard
}
```

In addition to generation 1 the following data elements are used:

**sensorGNSSSerialNumber** which identifies an external GNSS facility.

**rcmSerialNumber** which identifies a Remote Communication Module.

**sealDataCard** gives information about the seals that are attached to different components of the vehicle.



## 2.237. WorkshopCardHolderIdentification

Information, stored in a workshop card, related to the identification of the cardholder (Annex 1C requirement 311 and 334).

```
WorkshopCardHolderIdentification ::= SEQUENCE {
    workshopName           Name,
    workshopAddress        Address,
    cardHolderName         HolderName,
    cardHolderPreferredLanguage Language
}
```

**workshopName** is name of the workshop of the card holder.

**workshopAddress** is the address of the workshop of the card holder.

**cardHolderName** is the name and first name(s) of the holder (e.g. the name of the mechanic).

**cardHolderPreferredLanguage** is the preferred language of the card holder.

## 2.238. WorkshopCardPIN

Personal identification number of the Workshop Card (Annex 1C requirement 309 and 332).

```
WorkshopCardPIN ::= IA5String(SIZE(8))
```

**Value assignment:** The PIN known to the cardholder, right padded with 'FF' bytes up to 8 bytes.

## 2.239. W-VehicleCharacteristicConstant

Characteristic coefficient of the vehicle (definition k).

```
W-VehicleCharacteristicConstant ::= INTEGER(0..216-1)
```

**Value assignment:** Impulses per kilometer in the operating range 0 to 64 255 pulses/km.

## 2.240. VuPowerSupplyInterruptionRecord

Generation 2:

Information, stored in a vehicle unit, related to Power Supply Interruption events (Annex 1C requirement 117).

```
VuPowerSupplyInterruptionRecord ::= SEQUENCE {  
    eventType                EventFaultType,  
    eventRecordPurpose       EventFaultRecordPurpose,  
    eventBeginTime           TimeReal,  
    eventEndTime             TimeReal,  
    cardNumberAndGenDriverSlotBegin FullCardNumberAndGeneration,  
    cardNumberAndGenDriverSlotEnd   FullCardNumberAndGeneration,  
    cardNumberAndGenCodriverSlotBegin FullCardNumberAndGeneration,  
    cardNumberAndGenCodriverSlotEnd FullCardNumberAndGeneration,  
    similarEventsNumber       SimilarEventsNumber  
}
```

**eventType** is the type of the event.

**eventRecordPurpose** is the purpose for which this event has been recorded.

**eventBeginTime** is the date and time of beginning of event.

**eventEndTime** is the date and time of end of event.

**cardNumberAndGenDriverSlotBegin** identifies the card including its generation inserted in the driver slot at the beginning of the event.

**cardNumberAndGenDriverSlotEnd** identifies the card including its generation inserted in the driver slot at the end of the event.

**cardNumberAndGenCodriverSlotBegin** identifies the card including its generation inserted in the co-driver slot at the beginning of the event.

**cardNumberAndGenCodriverSlotEnd** identifies the card including its generation inserted in the co-driver slot at the end of the event.

**similarEventsNumber** is the number of similar events that day.

## 2.241. VuPowerSupplyInterruptionRecordArray

Generation 2:

Information, stored in a vehicle unit, related to Power Supply Interruption events (Annex 1C requirement 117).

```

VuPowerSupplyInterruptionRecordArray ::= SEQUENCE {
    recordType          RecordType,
    recordSize          INTEGER(1..65535),
    noOfRecords         INTEGER(0..65535),
    records              SET SIZE(noOfRecords) OF VuPowerSupplyInterruptionRecord
}

```

**recordType** denotes the type of the record (VuPowerSupplyInterruptionRecord). **Value Assignment:** See RecordType

**recordSize** is the size of the VuPowerSupplyInterruptionRecord in bytes.

**noOfRecords** is the number of records in the set records.

**records** is a set of power supply interruption events records.

#### 2.242. VuSensorExternalGNSSCoupledRecordArray

Generation 2:

A set of SensorExternalGNSSCoupledRecord plus metadata used in the download protocol.

```

VuSensorExternalGNSSCoupledRecordArray ::= SEQUENCE {
    recordType          RecordType,
    recordSize          INTEGER(1..65535),
    noOfRecords         INTEGER(0..65535),
    records              SET SIZE(noOfRecords) OF SensorExternalGNSSCoupledRecord
}

```

**recordType** denotes the type of the record (SensorExternalGNSSCoupledRecord). **Value Assignment:** See RecordType

**recordSize** is the size of the SensorExternalGNSSCoupledRecord in bytes.

**noOfRecords** is the number of records in the set records.

**records** is a set of Sensor External GNSS Coupled records.

#### 2.243. VuSensorPairedRecordArray

Generation 2:

A set of SensorPairedRecord plus metadata used in the download protocol.

VuSensorPairedRecordArray ::= SEQUENCE {

recordType	RecordType,
recordSize	INTEGER(1..65535),
noOfRecords	INTEGER(0..65535),
records	SET SIZE(noOfRecords) OF SensorPairedRecord

}

**recordType** denotes the type of the record (SensorPairedRecord). **Value Assignment:** See RecordType

**recordSize** is the size of the SensorPairedRecord in bytes.

**noOfRecords** is the number of records in the set records.

**records** is a set of sensor paired records.

### 3. Value and size range definitions

Definition of variable values used for definitions in paragraph 2.

TimeRealRange ::=  $2^{32}-1$

### 4. Character sets

IA5Strings use the ASCII characters as defined by ISO/IEC 8824-1. For readability and for easy referencing the value assignment is given below. The ISO/IEC 8824-1 supersedes this informative note in case of discrepancy.

! " # \$ % & ' ( ) \* + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ?  
@ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ] ^ \_  
` a b c d e f g h i j k l m n o p q r s t u v w x y z { | } ~

Other character strings (Address, Name, VehicleRegistrationNumber) use, in addition, characters from the decimal character code range 161 – 255 of the following 8-bit, standard character sets, specified by the Code Page number:

Standard Character Set	Code Page (Decimal)
ISO/IEC 8859-1 Latin-1 Western European	1
ISO/IEC 8859-2 Latin-2 Central European	2
ISO/IEC 8859-3 Latin-3 South European	3
ISO/IEC 8859-5 Latin / Cyrillic	5
ISO/IEC 8859-7 Latin / Greek	7
ISO/IEC 8859-9 Latin-5 Turkish	9
ISO/IEC 8859-13 Latin-7 Baltic Rim	13
ISO/IEC 8859-15 Latin-9	15
ISO/IEC 8859-16 Latin-10 South Eastern European	16
KOI8-R Latin / Cyrillic	80
KOI8-U Latin / Cyrillic	85

## 5. Encoding

When encoded with ASN.1 encoding rules, all data types defined shall be encoded according to ISO/IEC 8825-2, aligned variant.

## 6. Object Identifiers und Application Identifiers

### 6.1. Object Identifiers

The Object Identifiers (OIDs) listed in this chapter are only relevant for generation 2. These OIDs are specified in TR-03110-3 and repeated here for the sake of completeness. These OIDs are contained in the subtree of bsi-de:

```
bsi-de OBJECT IDENTIFIER ::= {
    itu-t(0) identified-organization(4) etsi(0)
    reserved(127) etsi-identified-organization(0) 7
}
```

**VU Authentication protocol identifiers**

id-TA OBJECT IDENTIFIER ::= {bsi-de protocols(2) smartcard(2) 2}

id-TA-ECDSA OBJECT IDENTIFIER ::= {id-TA 2}

id-TA-ECDSA-SHA-256 OBJECT IDENTIFIER ::= {id-TA-ECDSA 3}

id-TA-ECDSA-SHA-384 OBJECT IDENTIFIER ::= {id-TA-ECDSA 4}

id-TA-ECDSA-SHA-512 OBJECT IDENTIFIER ::= {id-TA-ECDSA 5}

Example: Suppose VU Authentication is to be done with SHA-384, then the object identifier to use is (in ASN.1 notation) bsi-de protocols(2) smartcard(2) 2 2 4. The value of this object identifier in dot notation is 0.4.0.127.0.7.2.2.2.2.4.

	Dot notation	Byte notation
id-TA-ECDSA-SHA-256	0.4.0.127.0.7.2.2.2.2.3	'04 00 7F 00 07 02 02 02 02 03'
id-TA-ECDSA-SHA-384	0.4.0.127.0.7.2.2.2.2.4	'04 00 7F 00 07 02 02 02 02 04'
id-TA-ECDSA-SHA-512	0.4.0.127.0.7.2.2.2.2.5	'04 00 7F 00 07 02 02 02 02 05'

**Chip Authentication protocol identifiers**

id-CA OBJECT IDENTIFIER ::= {bsi-de protocols(2) smartcard(2) 3}

id-CA-ECDH OBJECT IDENTIFIER ::= {id-CA 2}

id-CA-ECDH-AES-CBC-CMAC-128 OBJECT IDENTIFIER ::= {id-CA-ECDH 2}

id-CA-ECDH-AES-CBC-CMAC-192 OBJECT IDENTIFIER ::= {id-CA-ECDH 3}

id-CA-ECDH-AES-CBC-CMAC-256 OBJECT IDENTIFIER ::= {id-CA-ECDH 4}

Example: Suppose Chip Authentication is to be done by using the ECDH algorithm, resulting in an AES session key length of 128 bits. This session key will subsequently be used in the CBC mode of operation to ensure data confidentiality and with the CMAC algorithm to ensure data authenticity. Therefore, the object identifier to use is (in ASN.1 notation) bsi-de protocols(2) smartcard(2) 3 2 2. The value of this object identifier in dot notation is 0.4.0.127.0.7.2.2.3.2.2.

	Dot notation	Byte notation
id-CA-ECDH-AES-CBC-CMAC-128	0.4.0.127.0.7.2.2.3.2.2	'04 00 7F 00 07 02 02 03 02 02'
id-CA-ECDH-AES-CBC-CMAC-192	0.4.0.127.0.7.2.2.3.2.3	'04 00 7F 00 07 02 02 03 02 03'
id-CA-ECDH-AES-CBC-CMAC-256	0.4.0.127.0.7.2.2.3.2.4	'04 00 7F 00 07 02 02 03 02 04'

## 6.2. Application Identifiers

### Generation 2:

The Application Identifier (AID) for the External GNSS Facility (Generation 2) is given by 'FF 44 54 45 47 4D'. This is a proprietary AID according to ISO/IEC 7816-4.

Note: The last 5 bytes encode DTEGM for smart Tachograph External GNSS Facility.

The Application Identifier for the generation 2 tachograph card application is given by 'FF 53 4D 52 44 54'. This is a proprietary AID according to ISO/IEC 7816-4.

## **APPENDIX 2. TACHOGRAPH CARDS SPECIFICATION**

Date: 02-02-2016



---

**TABLE OF CONTENT**

<b>1. INTRODUCTION.....</b>	<b>227</b>
<b>1.1. Abbreviations _____</b>	<b>227</b>
<b>1.2. References _____</b>	<b>228</b>
<b>2. ELECTRICAL AND PHYSICAL CHARACTERISTICS .....</b>	<b>228</b>
<b>2.1. Supply Voltage and Current Consumption _____</b>	<b>229</b>
<b>2.2. Programming Voltage <math>V_{pp}</math> _____</b>	<b>229</b>
<b>2.3. Clock generation and Frequency _____</b>	<b>229</b>
<b>2.4. I/O Contact _____</b>	<b>229</b>
<b>2.5. States of the Card _____</b>	<b>229</b>
<b>3. HARDWARE AND COMMUNICATION.....</b>	<b>229</b>
<b>3.1. Introduction _____</b>	<b>229</b>
<b>3.2. Transmission Protocol _____</b>	<b>230</b>
3.2.1 .. Protocols	230
3.2.2 .. ATR	231
3.2.3 .. PTS	231
<b>3.3. Access Rules _____</b>	<b>232</b>
<b>3.4. Commands and error codes overview _____</b>	<b>235</b>
<b>3.5. Command descriptions _____</b>	<b>237</b>
3.5.1 ...SELECT	<b>Error! Bookmark not defined.</b> 240
3.5.2 .. Read Binary	<b>Error! Bookmark not defined.</b>
3.5.3 .. Update Binary	249
3.5.4 .. Get Challenge	256
3.5.5 .. Verify	258
3.5.6 .. Get Response	259
3.5.7 .. PSO: Verify Certificate	261
3.5.8 .. Internal Authenticate	262
3.5.9 .. External Authenticate	263
3.5.10. General Authenticate	264
3.5.11. Manage Security Environment	262
3.5.12. PSO: Hash	266
3.5.13. Perform Hash of File	267

3.5.14. PSO: Compute Digital Signature	268
3.5.15. PSO: Verify Digital Signature	269
3.5.16. Process DSRC Message	270
<b>4. TACHOGRAPH CARDS STRUCTURE</b>	<b>272</b>
<b>4.1. Master File MF</b>	<b>272</b>
<b>4.2. Driver card applications</b>	<b>274</b>
4.2.1 .. Driver card application generation 1	274
4.2.2 .. Driver card application generation 2	277
<b>4.3. Workshop card applications</b>	<b>282</b>
4.3.1 .. Workshop card application generation 1	282
4.3.2 .. Workshop card application generation 2	286
<b>4.4. Control card applications</b>	<b>292</b>
4.4.1 .. Control Card application generation 1	292
4.4.2 .. Control card application generation 2	294
<b>4.5. Company card applications</b>	<b>296</b>
4.5.1 .. Company card application generation 1	296
4.5.2 .. Company card application generation 2	297

## 1. Introduction

### 1.1. Abbreviations

For the purpose of this appendix, the following abbreviations apply.

<b>AC</b>	Access conditions
<b>AES</b>	Advanced Encryption Standard
<b>AID</b>	Application Identifier
<b>ALW</b>	Always
<b>APDU</b>	Application Protocol Data Unit (structure of a command)
<b>ATR</b>	Answer To Reset
<b>AUT</b>	Authenticated.
<b>C6, C7</b>	Contacts N° 6 and 7 of the card as described in ISO/IEC 7816-2
<b>cc</b>	clock cycles
<b>CHV</b>	Card holder Verification Information
<b>CLA</b>	Class byte of an APDU command
<b>DSRC</b>	Dedicated Short Range Communication
<b>DF</b>	Dedicated File. A DF can contain other files (EF or DF)
<b>ECC</b>	Elliptic Curve Cryptography
<b>EF</b>	Elementary File
<b>etu</b>	elementary time unit
<b>G1</b>	Generation 1
<b>G2</b>	Generation 2
<b>IC</b>	Integrated Circuit
<b>ICC</b>	Integrated Circuit Card
<b>ID</b>	Identifier
<b>IFD</b>	Interface Device
<b>IFS</b>	Information Field Size
<b>IFSC</b>	Information Field Size for the card
<b>IFSD</b>	Information Field Size Device (for the Terminal)
<b>INS</b>	Instruction byte of an APDU command
<b>Lc</b>	Length of the input data for a APDU command
<b>Le</b>	Length of the expected data (output data for a command)
<b>MF</b>	Master File (root DF)
<b>NAD</b>	Node Address used in T=1 protocol

<b>NEV</b>	Never
<b>P1-P2</b>	Parameter bytes
<b>PIN</b>	Personal Identification Number
<b>PRO SM</b>	Protected with secure messaging
<b>PTS</b>	Protocol Transmission Selection
<b>RFU</b>	Reserved for Future Use
<b>RST</b>	Reset (of the card)
<b>SFID</b>	Short EF Identifier
<b>SM</b>	Secure Messaging
<b>SW1-SW2</b>	Status bytes
<b>TS</b>	Initial ATR character
<b>VPP</b>	Programming Voltage
<b>VU</b>	Vehicle Unit
<b>XXh</b>	Value XX in hexadecimal notation
<b>'XXh'</b>	Value XX in hexadecimal notation
<b>  </b>	Concatenation symbol 03  04=0304

## 1.2. References

The following references are used in this Appendix:

ISO/IEC 7816-2	Identification cards - Integrated circuit cards - Part 2: Dimensions and location of the contacts. ISO/IEC 7816-2:2007.
ISO/IEC 7816-3	Identification cards - Integrated circuit cards - Part 3: Electrical interface and transmission protocols. ISO/IEC 7816-3:2006.
ISO/IEC 7816-4	Identification cards - Integrated circuit cards - Part 4: Organization, security and commands for interchange. ISO/IEC 7816-4:2013 + Cor 1: 2014.
ISO/IEC 7816-6	Identification cards - Integrated circuit cards - Part 6: Interindustry data elements for interchange. ISO/IEC 7816-6:2004 + Cor 1: 2006.
ISO/IEC 7816-8	Identification cards - Integrated circuit cards - Part 8: Commands for security operations. ISO/IEC 7816-8:2004.
ISO/IEC 9797-2	Information technology -- Security techniques -- Message Authentication Codes (MACs) -- Part 2: Mechanisms using a dedicated hash-function. ISO/IEC 9797-2:2011

## 2. Electrical and physical characteristics

TCS\_01 All electronic signals shall be in accordance with ISO/IEC 7816-3 unless specified otherwise.

TCS\_02 The location and dimensions of the card contacts shall comply with the ISO/IEC 7816-2.

### 2.1. Supply Voltage and Current Consumption

TCS\_03 The card shall work according to specifications within the consumption limits specified in ISO/IEC 7816-3.

TCS\_04 The card shall work with  $V_{cc} = 3V (\pm 0.3V)$  or with  $V_{cc} = 5V (\pm 0.5V)$ .

Voltage selection shall be performed according to ISO/IEC 7816-3.

### 2.2. Programming Voltage $V_{pp}$

TCS\_05 The card shall not require a programming voltage at pin C6. It is expected that pin C6 is not connected in an IFD. Contact C6 may be connected to  $V_{cc}$  in the card but shall not be connected to ground. This voltage should not be interpreted in any case.

### 2.3. Clock generation and Frequency

TCS\_06 The card shall operate within a frequency range of 1 to 5 MHz and may support higher frequencies. Within one card session the clock frequency may vary  $\pm 2\%$ . The clock frequency is generated by the Vehicle Unit and not the card itself. The duty cycle may vary between 40 and 60%.

TCS\_07 Under conditions contained into the card file EF ICC, the external clock can be stopped. The first byte of the EF ICC file body codes the Clockstop mode conditions:

<i>Low</i>		<i>High</i>		
Bit 3	Bit 2	Bit 1		
0	0	1		Clockstop allowed, no preferred level
0	1	1		Clockstop allowed, high level preferred
1	0	1		Clockstop allowed, low level preferred
0	0	0		Clockstop not allowed
0	1	0		Clockstop only allowed on high level
1	0	0		Clockstop only allowed on low level

Bits 4 to 8 are not used.

### 2.4. I/O Contact

TCS\_08 The I/O contact C7 is used to receive data from and to transmit data to the IFD. During operation only either the card or the IFD shall be in transmit mode. Should both units be in transmit mode no damage shall occur to the card. Unless transmitting, the card shall enter the reception mode.

### 2.5. States of the Card

TCS\_09 The card works in two states while the supply voltage is applied:

Operation state while executing commands or interfacing with Digital Unit,

Idle state at all other times; in this state all data shall be retained by the card.

## 3. Hardware and communication

### 3.1. Introduction

This paragraph describes the minimum functionality required by Tachograph cards and VUs to ensure correct operation and interoperability.

Tachograph cards are as compliant as possible with the available ISO/IEC applicable norms (especially ISO/IEC 7816). However, commands and protocols are fully described in order to specify some restricted usage or some differences if they exist. The commands specified are fully compliant with the referred norms except where indicated.

### 3.2. Transmission Protocol

TCS\_10 The Transmission protocol shall be compliant with ISO/IEC 7816-3 for T = 0 and T = 1. In particular, the VU shall recognise waiting time extensions sent by the card.

#### 3.2.1 Protocols

TCS\_11 The card shall provide both protocol **T=0** and protocol **T=1**. In addition the card may support further contact-oriented protocols.

TCS\_12 **T=0** is the default protocol, a **PTS** command is therefore necessary to change the protocol to **T=1**.

TCS\_13 Devices shall support **direct convention** in both protocols: the direct convention is hence mandatory for the card.

TCS\_14 The **Information Field Size Card** byte shall be presented at the ATR in character TA3. This value shall be at least 'F0h' (=240 bytes).

The following restrictions apply to the protocols:

TCS\_15 **T=0**

- The interface device shall support an answer on I/O after the rising edge of the signal on RST from 400 cc.
- The interface device shall be able to read characters separated with 12 etu.
- The interface device shall read an erroneous character and its repetition if separated with 13 etu. If an erroneous character is detected, the Error signal on I/O can occur between 1 etu and 2 etu. The device shall support a 1 etu delay.
- The interface device shall accept a 33 bytes ATR (TS+32)
- If TC1 is present in the ATR, the Extra Guard Time shall be present for characters sent by the interface device although characters sent by the card can still be separated with 12 etu. This is also true for the ACK character sent by the card after a P3 character emitted by the interface device.
- The interface device shall take into account a NUL character emitted by the card.
- The interface device shall accept the complementary mode for ACK.
- The get-response command cannot be used in chaining mode to get a data which length could exceed 255 bytes.

TCS\_16 **T=1**

- NAD byte : not used (NAD shall be set to '00').
- S-block ABORT : not used.
- S-block VPP state error : not used.
- The total chaining length for a data field will not exceed 255 bytes (to be ensured by the IFD).
- The Information Field Size Device (IFSD) shall be indicated by the IFD immediately after the ATR : the IFD shall transmit the S-Block IFS request after the ATR and the card shall send back S-Block IFS. The recommended value for IFSD is 254 bytes.

- The card will not ask for an IFS readjustment.

### 3.2.2 ATR

TCS\_17 The device checks ATR bytes, according to ISO/IEC 7816-3. No verification shall be done on ATR Historical Characters.

Example of Basic Biprotocol ATR according to ISO/IEC 7816-3

<i>Character</i>	<i>Value</i>	<i>Remarks</i>
TS	'3Bh'	Indicates direct convention.
T0	'85h'	TD1 present; 5 historical bytes are presents.
TD1	'80h'	TD2 present; T=0 to be used
TD2	'11h'	TA3 present; T=1 to be used
TA3	'XXh' (at least 'F0h')	Information Field Size Card ( IFSC)
TH1 to TH5	'XXh'	Historical characters
TCK	'XXh'	Check Character (exclusive OR)

TCS\_18 After the Answer To Reset (ATR), the Master File (MF) is implicitly selected and becomes the Current Directory.

### 3.2.3 PTS

TCS\_19 The default Protocol is T=0. To set the T=1 protocol, a PTS (also known as PPS) must be sent to the card by the device.

TCS\_20 As both T=0 and T=1 protocols are mandatory for the card, the basic PTS for protocol switching is mandatory for the card.

The PTS can be used, as indicated in ISO/IEC 7816-3, to switch to higher baud rates than the default one proposed by the card in the ATR if any (TA(1) byte).

Higher baud rates are optional for the card.

TCS\_21 If no other baud rate than the default one are supported (or if the selected baud rate is not supported), the card shall respond to the PTS correctly according to ISO/IEC 7816-3 by omitting the PPS1 byte.

Examples of basic PTS for protocol selection are the following:

<i>Character</i>	<i>Value</i>	<i>Remarks</i>
PPSS	'FFh'	The Initiate Character.
PPS0	'00h' or '01h'	PPS1 to PPS3 are not present; '00h' to select T0, '01h' to select T1.
PK	'XXh'	Check Character: 'XXh' = 'FFh' if PPS0 = '00h', 'XXh' = 'FEh' if PPS0 = '01h'.

## 3.3. Access Rules

TCS\_22 An access rule specifies for an access mode, i.e. command, the corresponding security conditions. If these security conditions are fulfilled the corresponding command is processed.

TCS\_23 The following security conditions are used for the tachograph card:

<i>Abbreviation</i>	<i>Meaning</i>
<b>ALW</b>	The action is always possible and can be executed without any restriction. Command and response APDU are sent in plain text, i.e. without secure messaging.
<b>NEV</b>	The action is never possible.
<b>PLAIN-C</b>	The command APDU is sent in plain, i.e. without secure messaging.
<b>PWD</b>	The action may only be executed if the workshop card PIN has been successfully verified, i.e. if the card internal security status "PIN_Verified" is set. The command must be sent without secure messaging.
<b>EXT-AUT-G1</b>	The action may only be executed if the External Authenticate command for the generation 1 authentication (see also Appendix 11 Part A) has been successfully performed.
<b>SM-MAC-G1</b>	The APDU (command and response) must be applied with generation 1 secure messaging in authentication-only mode (see Appendix 11 Part A).
<b>SM-C-MAC-G1</b>	The command APDU must be applied with generation 1 secure messaging in authentication only mode (see Appendix 11 Part A).
<b>SM-R-ENC-G1</b>	The response APDU must be applied with generation 1 secure messaging in encryption mode (see Appendix 11 Part A), i.e. no message authentication code is returned.
<b>SM-R-ENC-MAC-G1</b>	The response APDU must be applied with generation 1 secure messaging in encrypt-then-authenticate mode (see Appendix 11 Part A).
<b>SM-MAC-G2</b>	The APDU (command and response) must be applied with generation 2 secure messaging in authentication-only mode (see Appendix 11 Part B).
<b>SM-C-MAC-G2</b>	The command APDU must be applied with generation 2 secure messaging in authentication only mode (see Appendix 11 Part B).
<b>SM-R-ENC-MAC-G2</b>	The response APDU must be applied with generation 2 secure messaging in encrypt-then-authenticate mode (see Appendix 11 Part B).

TCS\_24 These security conditions can be linked in the following ways:

- **AND** : All security conditions must be fulfilled
- **OR** : At least one security condition must be fulfilled



The access rules for the file system, i.e. the SELECT, READ BINARY and UPDATE BINARY command, are specified in chapter 4. The access rules for the remaining commands are specified in the following tables.

TCS\_25 In the DF Tachograph G1 application the following access rules are used:

<i>Command</i>	<i>Driver Card</i>	<i>Workshop Card</i>	<i>Control Card</i>	<i>Company Card</i>
External Authenticate				
• For generation 1 authentication	ALW	ALW	ALW	ALW
• For generation 2 authentication	ALW	PWD	ALW	ALW
Internal Authenticate	ALW	PWD	ALW	ALW
General Authenticate	ALW	ALW	ALW	ALW
Get Challenge	ALW	ALW	ALW	ALW
MSE:SET AT	ALW	ALW	ALW	ALW
MSE:SET DST	ALW	ALW	ALW	ALW
Process DSRC Message	Not applicable	Not applicable	Not applicable	Not applicable
PSO: Compute Digital Signature	ALW OR SM-MAC-G2	ALW OR SM-MAC-G2	Not applicable	Not applicable
PSO: Hash	Not applicable	Not applicable	ALW	Not applicable
PSO: Hash of File	ALW OR SM-MAC-G2	ALW OR SM-MAC-G2	Not applicable	Not applicable
PSO: Verify Certificate	ALW	ALW	ALW	ALW
PSO: Verify Digital Signature	Not applicable	Not applicable	ALW	Not applicable
Verify	Not applicable	ALW	Not applicable	Not applicable

TCS\_26 In the DF Tachograph\_G2 application the following access rules are used:

<i>Command</i>	<i>Driver Card</i>	<i>Workshop Card</i>	<i>Control Card</i>	<i>Company Card</i>
External Authenticate				
• For generation 1 authentication	Not applicable	Not applicable	Not applicable	Not applicable
• For generation 2 authentication	ALW	PWD	ALW	ALW
Internal Authenticate	Not applicable	Not applicable	Not applicable	Not applicable
General Authenticate	ALW	ALW	ALW	ALW
Get Challenge	ALW	ALW	ALW	ALW
MSE:SET AT	ALW	ALW	ALW	ALW
MSE:SET DST	ALW	ALW	ALW	ALW
Process DSRC Message	Not applicable	ALW	ALW	Not applicable
PSO: Compute Digital Signature	ALW OR SM-MAC-G2	ALW OR SM-MAC-G2	Not applicable	Not applicable
PSO: Hash	Not applicable	Not applicable	ALW	Not applicable
PSO: Hash of File	ALW OR SM-MAC-G2	ALW OR SM-MAC-G2	Not applicable	Not applicable
PSO: Verify Certificate	ALW	ALW	ALW	ALW
PSO: Verify Digital Signature	Not applicable	Not applicable	ALW	Not applicable
Verify	Not applicable	ALW	Not applicable	Not applicable

TCS\_27 In the MF the following access rules are used:

<i>Command</i>	<i>Driver Card</i>	<i>Workshop Card</i>	<i>Control Card</i>	<i>Company Card</i>
External Authenticate				
• For generation 1 authentication	Not applicable	Not applicable	Not applicable	Not applicable

<i>Command</i>	<i>Driver Card</i>	<i>Workshop Card</i>	<i>Control Card</i>	<i>Company Card</i>
• For generation 2 authentication	ALW	PWD	ALW	ALW
Internal Authenticate	Not applicable	Not applicable	Not applicable	Not applicable
General Authenticate	ALW	ALW	ALW	ALW
Get Challenge	ALW	ALW	ALW	ALW
MSE:SET AT	ALW	ALW	ALW	ALW
MSE:SET DST	ALW	ALW	ALW	ALW
Process DSRC Message	Not applicable	Not applicable	Not applicable	Not applicable
PSO: Compute Digital Signature	Not applicable	Not applicable	Not applicable	Not applicable
PSO: Hash	Not applicable	Not applicable	Not applicable	Not applicable
PSO: Hash of File	Not applicable	Not applicable	Not applicable	Not applicable
PSO: Verify Certificate	ALW	ALW	ALW	ALW
Verify	Not applicable	ALW	Not applicable	Not applicable

TCS\_28 A tachograph card may or may not accept a command with a higher level of security than the one specified in the security conditions. I.e. if the security condition is ALW (or PLAIN-C) the card may accept a command with secure messaging (encryption and / or authentication mode). If the security condition requires secure messaging with authentication mode, the tachograph card may accept a command with secure messaging of the same generation in authentication and encryption mode.

Note: The command descriptions provide more information on the support of the commands for the different tachograph card types and the different DFs.

### 3.4. Commands and error codes overview

Commands and file organisation are deduced from and complies with ISO/IEC 7816-4.

This section describes the following APDU command-response pairs. The command variants which are supported by a generation 1 and 2 application are specified in the corresponding command descriptions.

<i>Command</i>	<i>INS</i>
SELECT	'A4h'
READ BINARY	'B0h', 'B1h'
UPDATE BINARY	'D6h', 'D7h'
GET CHALLENGE	'84h'
VERIFY	'20h'
GET RESPONSE	'C0h'
PERFORM SECURITY OPERATION	'2Ah'
• VERIFY CERTIFICATE	
• COMPUTE DIGITAL SIGNATURE	
• VERIFY DIGITAL SIGNATURE	
• HASH	
• PERFORM HASH OF FILE	
• PROCESS DSRC MESSAGE	
INTERNAL AUTHENTICATE	'88h'
EXTERNAL AUTHENTICATE	'82h'
MANAGE SECURITY ENVIRONMENT	'22h'
• SET DIGITAL SIGNATURE TEMPLATE	
• SET AUTHENTICATION TEMPLATE	
GENERAL AUTHENTICATE	'86h'

TCS\_29 The status words SW1 SW2 are returned in any response message and denote the processing state of the command.

<i>SW1</i>	<i>SW2</i>	<i>Meaning</i>
90	00	Normal processing.
61	XX	Normal processing. XX = number of response bytes available.
62	81	Warning processing. Part of returned data may be corrupted
63	00	Authentication failed (Warning)
63	CX	Wrong CHV (PIN). Remaining attempts counter provided by 'X'.
64	00	Execution error - State of non-volatile memory unchanged. Integrity error.
65	00	Execution error - State of non-volatile memory changed

<i>SW1</i>	<i>SW2</i>	<i>Meaning</i>
65	81	Execution error - State of non-volatile memory changed – Memory failure
66	88	Security error : wrong cryptographic checksum (during Secure Messaging) or wrong certificate (during certificate verification) or wrong cryptogram (during external authentication) or wrong signature (during signature verification)
67	00	Wrong length (wrong Lc or Le)
68	82	Secure messaging not supported
68	83	Last command of the chain expected
69	00	Forbidden command (no response available in T=0)
69	82	Security status not satisfied.
69	83	Authentication method blocked.
69	85	Conditions of use not satisfied.
69	86	Command not allowed (no current EF).
69	87	Expected Secure Messaging Data Objects missing
69	88	Incorrect Secure Messaging Data Objects
6A	80	Incorrect parameters in data field
6A	82	File not found.
6A	86	Wrong parameters P1-P2.
6A	88	Referenced data not found.
6B	00	Wrong parameters (offset outside the EF).
6C	XX	Wrong length, SW2 indicates the exact length. No data field is returned.
6D	00	Instruction code not supported or invalid.
6E	00	Class not supported.
6F	00	Other checking errors

TCS\_30 If more than one error condition is fulfilled in one command APDU the card may return any of the appropriate status words.

### 3.5. Command descriptions

The mandatory commands for the Tachograph cards are described in this chapter.

Additional relevant details, related to cryptographic operations involved, are given in Appendix 11 Common security mechanisms for Tachograph Generation 1 and Generation 2.

All commands are described independently of the used protocol (T=0 or T=1). The APDU bytes CLA, INS, P1, P2, Lc and Le are always indicated. If Lc or Le is not needed for the described command, the associated length, value and description are empty.

- TCS\_31 If both length bytes (Lc and Le) are requested, the described command has to be split in two parts if the IFD is using protocol T=0 : the IFD sends the command as described with P3=Lc + data and then sends a GET RESPONSE (see § **Error! Reference source not found.**) command with P3=Le.
- TCS\_32 If both length bytes are requested, and Le=0 (secure messaging):
- When using protocol T=1, the card shall answer to Le=0 by sending all available output data.
  - When using protocol T=0, the IFD shall send the first command with P3=Lc + data, the card shall answer (to this implicit Le=0) by the Status bytes '61La', where La is the number of response bytes available. The IFD shall then generate a GET RESPONSE command with P3 = La to read the data.
- TCS\_33 A tachograph card may support extended length fields according to ISO/IEC 7816-4 as an optional feature. A tachograph card that supports extended length fields shall
- Indicate the extended length field support in the ATR
  - Provide the supported buffer sizes by means of the extended length information in the EF ATR/INFO see TCS\_146.
  - Indicate whether it supports extended length fields for T = 1 and / or T = 0 in the EF Extended Length, see TCS\_147.
  - Support extended length fields for the tachograph application generation 1 and 2.

Notes:

All commands are specified for short length fields. The usage of extended length APDUs is clear from ISO/IEC 7816-4.

In general the commands are specified for the plain mode, i.e. without secure messaging, as the secure messaging layer is specified in Appendix 11. It is clear from the access rules for a command whether the command shall support secure messaging or not and whether the command shall support generation 1 and / or generation 2 secure messaging. Some command variants are described with secure messaging to illustrate the usage of secure messaging.

- TCS\_34 The VU shall perform the complete generation 2 VU – card mutual authentication protocol for a session including the certificate verification (if required) either in the DF Tachograph, the DF Tachograph\_G2 or the MF.

### 3.5.1 SELECT

This command is compliant with ISO/IEC 7816-4, but has a restricted usage compared to the command defined in the norm.

The SELECT command is used:

- to select an application DF (selection by name must be used)
- to select an elementary file corresponding to the submitted file ID

#### Selection by name (AID)

This command allows selecting an application DF in the card.

TCS\_35 This command can be performed from anywhere in the file structure (after the ATR or at any time).

TCS\_36 The selection of an application resets the current security environment. After performing the application selection, no current public key is selected anymore. The EXT-AUT-G1 access condition is also lost. If the command was performed without secure messaging, the former secure messaging session keys are no longer available.

#### TCS\_37 Command Message

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
CLA	1	'00h'	
INS	1	'A4h'	
P1	1	'04h'	Selection by name (AID)
P2	1	'0Ch'	No response expected
Lc	1	'NNh'	Number of bytes sent to the card (length of the AID) : '06h' for the Tachograph application
#6- #(5+N N)	NN	'XX..X Xh'	AID : 'FF 54 41 43 48 4F' for the Generation 1 tachograph application AID : 'FF 53 4D 52 44 54' for the Generation 2 tachograph application

No response to the SELECT command is needed (Le absent in T=1, or no response asked in T=0).

#### TCS\_38 Response Message (no response asked)

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
S W	2	'XXXXh'	Status Words (SW1,SW2)

- ◆ If the command is successful, the card returns '9000'.
- ◆ If the application corresponding with the AID is not found, the processing state returned is '6A82'.
- ◆ In T=1, if the byte Le is present, the state returned is '6700'.
- ◆ In T=0, if a response is asked after the SELECT command, the state returned is '6900'.
- ◆ If the selected application is considered corrupted (integrity error is detected within the file attributes), the processing state returned is '6400' or '6581'.

Selection of an Elementary File using its File Identifier

TCS\_39 **Command Message**

TCS\_40 A tachograph card shall support the generation 2 secure messaging as specified in Appendix 11 Part B for this command variant.

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
CLA	1	'00h'	
INS	1	'A4h'	
P1	1	'02h'	Selection of an EF under the current DF
P2	1	'0Ch'	No response expected
Lc	1	'02h'	Number of bytes sent to the card
#6- #7	2	'XXXXh'	File Identifier

No response to the SELECT command is needed (Le absent in T=1, or no response asked in T=0).

TCS\_41 **Response Message (no response asked)**

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
SW	2	'XXXXh'	Status Words (SW1,SW2)

- If the command is successful, the card returns '**9000**'.
- If the file corresponding with the file identifier is not found, the processing state returned is '**6A82**'.
- In T=1, if the byte Le is present, the state returned is '**6700**'.
- In T=0, if a response is asked after the SELECT command, the state returned is '**6900**'.
- If the selected file is considered corrupted (integrity error is detected within the file attributes), the processing state returned is '**6400**' or '**6581**'.

### 3.5.2 READ BINARY

This command is compliant with ISO/IEC 7816-4, but has a restricted usage compared to the command defined in the norm.

The READ BINARY command is used to read data from a transparent file.

The response of the card consists of returning the data read, optionally encapsulated in a secure messaging structure.

Command with offset in P1-P2

This command enables the IFD to read data from the EF currently selected, without secure messaging.



Note: This command without secure messaging can only be used to read a file that supports the ALW security condition for the Read access mode.

#### TCS\_42 Command Message

Byte	Length	Value	Description
CLA	1	'00h'	
INS	1	'B0h'	Read Binary
P1	1	'XXh'	Offset in bytes from the beginning of the file : Most Significant Byte
P2	1	'XXh'	Offset in bytes from the beginning of the file : Least Significant Byte
Le	1	'XXh'	Length of data expected. Number of Bytes to be read.

Note: bit 8 of P1 must be set to 0.

#### TCS\_43 Response Message

Byte	Length	Value	Description
#1- #X	X	'XX..X Xh'	Data read
SW	2	'XXXX h'	Status Words (SW1,SW2)

- ◆ If the command is successful, the card returns '**9000**'.
- ◆ If no EF is selected, the processing state returned is '**6986**'.
- ◆ If the security conditions of the selected file are not satisfied, the command is interrupted with '**6982**'.
- ◆ If the Offset is not compatible with the size of the EF (Offset > EF size), the processing state returned is '**6B00**'.
- ◆ If the size of the data to be read is not compatible with the size of the EF (Offset + Le > EF size) the processing state returned is '**6700**' or '**6Cxx**' where 'xx' indicates the exact length.
- ◆ If an integrity error is detected within the file attributes, the card shall consider the file as corrupted and unrecoverable, the processing state returned is '**6400**' or '**6581**'.
- ◆ If an integrity error is detected within the stored data, the card shall return the demanded data, and the processing state returned is '**6281**'.

Command with secure messaging (examples)

This command enables the IFD to read data from the EF currently selected with secure messaging, in order to verify the integrity of the data received and to protect the confidentiality of the data if the security condition SM-R-ENC-MAC-G1 (generation 1) or SM-R-ENC-MAC-G2 (generation 2) is applied.

#### TCS\_44 Command Message

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
CLA	1	'0Ch'	Secure Messaging asked
INS	1	'B0h'	Read Binary
P1	1	'XXh'	P1 ( offset in bytes from the beginning of the file) : Most Significant Byte
P2	1	'XXh'	P2 ( offset in bytes from the beginning of the file) : Least Significant Byte
Lc	1	'XXh'	Length of input data for secure messaging
#6	1	'97h'	T <sub>LE</sub> : Tag for expected length specification.
#7	1	'01h'	L <sub>LE</sub> : Length of expected length
#8	1	'NNh'	Expected length specification (original Le) : Number of Bytes to be read
#9	1	'8Eh'	T <sub>CC</sub> : Tag for cryptographic checksum
#10	1	'XXh'	L <sub>CC</sub> : Length of following cryptographic checksum '04h' for Generation 1 secure messaging (see Appendix 11 Part A) '08h', '0Ch' or '10h' depending on AES key length for Generation 2 secure messaging (see Appendix 11 Part B)
#11- #(10+ L)	L	'XX.. XXh'	Cryptographic checksum
Le	1	'00h'	As specified in ISO/IEC 7816-4

TCS\_45 **Response Message if SM-R-ENC-MAC-G1 (generation 1) / SM-R-ENC-MAC-G2 (generation 2) is not required and if Secure Messaging input format is correct:**

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
#1	1	'99h'	Tag for Processing Status (SW1-SW2) – optional for generation 1 secure messaging
#2	1	'02h'	Length of Processing Status
#3 - #4	2	'XX XXh'	Processing Status of the unprotected response APDU
#5	1	'81h'	T <sub>PV</sub> : Tag for plain value data
#6	L	'NNh' or '81	L <sub>PV</sub> : length of returned data (=original Le). L is 2 bytes if L <sub>PV</sub> >127 bytes.

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
		NNh'	
#(6+L)- #(5+L+NN)	N N	'XX..X Xh'	Plain Data value
#(6+L+NN)	1	'8Eh'	T <sub>CC</sub> : Tag for cryptographic checksum
#(7+L+NN)	1	'XXh'	L <sub>CC</sub> : Length of following cryptographic checksum  '04h' for Generation 1 secure messaging (see Appendix 11 Part A)  '08h', '0Ch' or '10h' depending on AES key length for Generation 2 secure messaging (see Appendix 11 Part B)
#(8+L+NN)- #(7+M+L+NN)	M	'XX..X Xh'	Cryptographic checksum
SW	2	'XXXX h'	Status Words (SW1,SW2)

TCS\_46 **Response Message if SM-R-ENC-MAC-G1 (generation 1) / SM-R-ENC-MAC-G2 (generation 2) is required and if Secure Messaging input format is correct:**

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
#1	1	'87h'	T <sub>PICG</sub> : Tag for encrypted data (cryptogram)
#2	L	'MMh' or '81 MMh'	L <sub>PICG</sub> : length of returned encrypted data (different of original Le of the command due to padding).  L is 2 bytes if L <sub>PICG</sub> > 127 bytes.
#(2+L)- #(1+L+MM)	M M	'01XX.. XXh'	Encrypted Data : Padding Indicator and cryptogram
#(2+L+MM)	1	'99h'	Tag for Processing Status (SW1-SW2) – optional for generation 1 secure messaging
#(3+L+MM)	1	'02h'	Length of Processing Status
#(4+L+MM) - #(5+L+MM)	2	'XX XXh'	Processing Status of the unprotected response APDU
#(6+L+MM)	1	'8Eh'	T <sub>CC</sub> : Tag for cryptographic checksum
#(7+L+MM)	1	'XXh'	L <sub>CC</sub> : Length of following cryptographic checksum  '04h' for Generation 1 secure messaging

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
			(see Appendix 11 Part A)
			'08h', '0Ch' or '10h' depending on AES key length for Generation 2 secure messaging (see Appendix 11 Part B)
#(8+L+MM)- #(7+N+L+MM)	N	'XX..X Xh'	Cryptographic checksum
SW	2	'XXXX h'	Status Words (SW1,SW2)

The READ BINARY command may return regular processing states listed in TCS\_43 under Tag '99h' as described in TCS\_59 using the secure messaging response structure.

Additionally, some errors specifically related to secure messaging can happen. In that case, the processing state is simply returned, with no secure messaging structure involved:

#### TCS\_47 Response Message if incorrect Secure Messaging input format

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
SW	2	'XX XXh'	Status Words (SW1,SW2)

- ◆ If no current session key is available, the processing state '**6A88**' is returned. It happens either if the session key has not already been generated or if the session key validity has expired (in this case the IFD must re-run a mutual authentication process to set a new session key).
- ◆ If some expected data objects (as specified above) are missing in the secure messaging format, the processing state '**6987**' is returned : this error happens if an expected tag is missing or if the command body is not properly constructed.
- ◆ If some data objects are incorrect, the processing state returned is '**6988**': this error happens if all the required tags are present but some lengths are different from the ones expected.
- ◆ If the verification of the cryptographic checksum fails, the processing state returned is '**6688**'.

Command with short EF (Elementary File) identifier

This command variant enables the IFD to select an EF by means of a short EF identifier and read data from this EF.

TCS\_48 A tachograph card shall support this command variant for all Elementary Files with a specified short EF identifier. These short EF identifiers are specified in chapter 4.

#### TCS\_49 Command Message

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
CLA	1	'00h'	
INS	1	'B0h'	Read Binary

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
P1	1	'XXh'	Bit 8 is set to 1 Bit 7 and 6 are set to 00 Bit 5 – 1 encode the short EF identifier of the corresponding EF
P2	1	'XXh'	Encodes an offset from 0 to 255 bytes in the EF referenced by P1
Le	1	'XXh'	Length of data expected. Number of Bytes to be read.

Note: The short EF identifiers used for the Generation 2 tachograph application are specified in chapter 4.

If P1 encodes a short EF identifier and the command is successful, the identified EF becomes the currently selected EF (current EF).

#### TCS\_50 Response Message

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
#1- #L	L	'XX...X Xh'	Data read
SW	2	'XXX Xh'	Status Words (SW1,SW2)

- ◆ If the command is successful, the card returns '**9000**'.
- ◆ If the file corresponding with the short EF identifier is not found, the processing state returned is '**6A82**'.
- ◆ If the security conditions of the selected file are not satisfied, the command is interrupted with '**6982**'.
- ◆ If the Offset is not compatible with the size of the EF (Offset > EF size), the processing state returned is '**6B00**'.
- ◆ If the size of the data to be read is not compatible with the size of the EF (Offset + Le > EF size) the processing state returned is '**6700**' or '**6Cxx**' where 'xx' indicates the exact length.
- ◆ If an integrity error is detected within the file attributes, the card shall consider the file as corrupted and unrecoverable, the processing state returned is '**6400**' or '**6581**'.
- ◆ If an integrity error is detected within the stored data, the card shall return the demanded data, and the processing state returned is '**6281**'.

Command with odd instruction byte

This command variant enables the IFD to read data from an EF with 32768 bytes or more.

TCS\_51 A tachograph card which supports EFs with 32768 bytes or more shall support this command variant for these EFs. A tachograph card may or may not support this command variant for other EFs with the exception of the EF Sensor\_Installation\_Data see TCS\_156 and TCS\_160.

#### TCS\_52 Command Message

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
-------------	---------------	--------------	--------------------

Byte	Length	Value	Description
CLA	1	'00h'	
INS	1	'B1h'	Read Binary
P1	1	'00h'	Current EF
P2	1	'00h'	
Lc	1	'NNh'	Lc Length of offset data object.
#6- #(5+NN )	NN	'XX..X Xh'	Offset data object: Tag '54h' Length '01h' or '02h' Value offset
Le	1	'XXh'	Number of Bytes to be read.

The IFD shall encode the offset data object's length with a minimum possible number of octets, i.e. using the length byte '01h' the IFD shall encode an offset from 0 to 255 and using the length byte '02h' an offset from '256' up to '65535' bytes.

#### TCS\_53 Response Message

Byte	Length	Value	Description
#1- #L	L	'XX..X Xh'	Data read encapsulated in a discretionary data object with tag '53h'.
SW	2	'XXX Xh'	Status Words (SW1,SW2)

- ◆ If the command is successful, the card returns '**9000**'.
- ◆ If no EF is selected, the processing state returned is '**6986**'.
- ◆ If the security conditions of the selected file are not satisfied, the command is interrupted with '**6982**'.
- ◆ If the Offset is not compatible with the size of the EF (Offset > EF size), the processing state returned is '**6B00**'.
- ◆ If the size of the data to be read is not compatible with the size of the EF (Offset + Le > EF size) the processing state returned is '**6700**' or '**6Cxx**' where 'xx' indicates the exact length.
- ◆ If an integrity error is detected within the file attributes, the card shall consider the file as corrupted and unrecoverable, the processing state returned is '**6400**' or '**6581**'.
- ◆ If an integrity error is detected within the stored data, the card shall return the demanded data, and the processing state returned is '**6281**'.

Command with secure messaging (example)

The following example illustrates the usage of secure messaging if the security condition SM-MAC-G2 applies.

#### TCS\_54 Command message

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
CLA	1	'0Ch'	Secure Messaging asked
INS	1	'B1h'	Read Binary
P1	1	'00h'	Current EF
P2	1	'00h'	
Lc	1	'XXh'	Length of the secured data field
#6	1	'B3h'	Tag for plain value data encoded in BER-TLV
#7	1	'NNh'	L <sub>PV</sub> : length of transmitted data
#(8)-#(7+NN)	N	'XX..X Xh'	Plain Data encoded in BER-TLV, i.e. the offset data object with tag '54'
#(8+NN)	1	'97h'	T <sub>LE</sub> : Tag for expected length specification.
#(9+NN)	1	'01h'	L <sub>LE</sub> : Length of expected length
#(10+NN)	1	'XXh'	Expected length specification (original Le): Number of bytes to be read
#(11+NN)	1	'8Eh'	T <sub>CC</sub> : Tag for cryptographic checksum
#(12+NN)	1	'XXh'	L <sub>CC</sub> : Length of following cryptographic checksum  '08h', '0Ch' or '10h' depending on AES key length for Generation 2 secure messaging (see Appendix 11 Part B)
#(13+NN)- #(12+M+NN)	M	'XX..X Xh'	Cryptographic checksum
Le	1	'00h'	As specified in ISO/IEC 7816-4

TCS\_55 Response message if the command is successful

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
#1	1	'B3h'	Plain Data encoded in BER-TLV
#2	L	'NNh' or '81 NNh'	L <sub>PV</sub> : length of returned data (=original Le). L is 2 bytes if L <sub>PV</sub> >127 bytes.
#(2+L)- #(1+L+NN)	N	'XX..X Xh'	Plain Data value encoded in BER-TLV, i.e. data read encapsulated in a discretionary

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
			data object with tag '53h'.
#(2+L+NN)	1	'99h'	Processing Status of the unprotected response APDU
#(3+L+NN)	1	'02h'	Length of Processing Status
#(4+L+NN) - #(5+L+NN)	2	'XX XXh'	Processing Status of the unprotected response APDU
#(6+L+NN)	1	'8Eh'	T <sub>CC</sub> : Tag for cryptographic checksum
#(7+L+NN)	1	'XXh'	L <sub>CC</sub> : Length of following cryptographic checksum  '08h', '0Ch' or '10h' depending on AES key length for Generation 2 secure messaging (see Appendix 11 Part B)
#(8+L+NN)- #(7+M+L+NN)	M	'XX..X Xh'	Cryptographic checksum
SW	2	'XXXX h'	Status Words (SW1,SW2)

### 3.5.3 UPDATE BINARY

This command is compliant with ISO/IEC 7816-4, but has a restricted usage compared to the command defined in the norm.

The UPDATE BINARY command message initiates the update (erase + write) of the bits already present in an EF binary with the bits given in the command APDU.

Command with offset in P1-P2

This command enables the IFD to write data into the EF currently selected, without the card verifying the integrity of data received.

Note: This command without secure messaging can only be used to update a file that supports the ALW security condition for the Update access mode.

#### TCS\_56 Command Message

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
CLA	1	'00h'	
INS	1	'D6h'	Update Binary
P1	1	'XXh'	Offset in bytes from the beginning of the file : Most Significant Byte
P2	1	'XXh'	Offset in bytes from the beginning of the file : Least



<i>Byte</i>	<i>Length</i> <i>h</i>	<i>Value</i>	<i>Description</i>
			Significant Byte
Lc	1	'NNh'	Lc Length of data to Update. Number of bytes to be written.
#6- #(5+NN )	NN	'XX..X Xh'	Data to be written

Note: bit 8 of P1 must be set to 0.

#### TCS\_57 Response Message

<i>Byte</i>	<i>Length</i> <i>h</i>	<i>Value</i>	<i>Description</i>
SW	2	'XXX Xh'	Status Words (SW1,SW2)

- ◆ If the command is successful, the card returns '9000'.
- ◆ If no EF is selected, the processing state returned is '6986'.
- ◆ If the security conditions of the selected file are not satisfied, the command is interrupted with '6982'.
- ◆ If the Offset is not compatible with the size of the EF (Offset > EF size), the processing state returned is '6B00'.
- ◆ If the size of the data to be written is not compatible with the size of the EF (Offset + Lc > EF size) the processing state returned is '6700'.
- ◆ If an integrity error is detected within the file attributes, the card shall consider the file as corrupted and unrecoverable, the processing state returned is '6400' or '6500'.
- ◆ If writing is unsuccessful, the processing state returned is '6581'.

Command with secure messaging (examples)

This command enables the IFD to write data into the EF currently selected, with the card verifying the integrity of data received. As no confidentiality is required, the data are not encrypted.

#### TCS\_58 Command Message

<i>Byte</i>	<i>Length</i> <i>Len</i> <i>gth</i>	<i>Value</i>	<i>Description</i>
CLA	1	'0Ch'	Secure Messaging asked
INS	1	'D6h'	Update Binary
P1	1	'XXh'	Offset in bytes from the beginning of the file : Most Significant Byte
P2	1	'XXh'	Offset in bytes from the beginning of the file :

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
			Least Significant Byte
Lc	1	'XXh'	Length of the secured data field
#6	1	'81h'	T <sub>PV</sub> : Tag for plain value data
#7	L	'NNh' or '81 NNh'	L <sub>PV</sub> : length of transmitted data. L is 2 bytes if L <sub>PV</sub> > 127 bytes.
#(7+L)-#(6+L+NN)	N N	'XX..X Xh'	Plain Data value (Data to be written)
#(7+L+NN)	1	'8Eh'	T <sub>CC</sub> : Tag for cryptographic checksum
#(8+L+NN)	1	'XXh'	L <sub>CC</sub> : Length of following cryptographic checksum '04h' for Generation 1 secure messaging (see Appendix 11 Part A) '08h', '0Ch' or '10h' depending on AES key length for Generation 2 secure messaging (see Appendix 11 Part B)
#(9+L+NN)- #(8+M+L+NN)	M	'XX..X Xh'	Cryptographic checksum
Le	1	'00h'	As specified in ISO/IEC 7816-4

TCS\_59 **Response message if correct Secure Messaging input format**

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
#1	1	'99h'	T <sub>SW</sub> : Tag for Status Words (to be protected by CC)
#2	1	'02h'	L <sub>SW</sub> : length of returned Status Words
#3- #4	2	'XXX Xh'	Processing Status of the unprotected response APDU
#5	1	'8Eh'	T <sub>CC</sub> : Tag for cryptographic checksum
#6	1	'XXh'	L <sub>CC</sub> : Length of following cryptographic checksum '04h' for Generation 1 secure messaging (see Appendix 11 Part A) '08h', '0Ch' or '10h' depending on AES key length for Generation 2 secure messaging (see Appendix 11 Part B)
#7- #(6+	L	'XX..X Xh'	Cryptographic checksum

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
L)			
SW	2	'XXX Xh'	Status Words (SW1,SW2)

The "regular" processing states, described for the UPDATE BINARY command with no secure messaging (see §0), can be returned using the response message structure described above.

Additionally, some errors specifically related to secure messaging can happen. In that case, the processing state is simply returned, with no secure messaging structure involved:

#### TCS\_60 Response Message if error in secure messaging

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
SW	2	'XX XXh'	Status Words (SW1,SW2)

- ◆ If no current session key is available, the processing state '6A88' is returned.
- ◆ If some expected data objects (as specified above) are missing in the secure messaging format, the processing state '6987' is returned : this error happens if an expected tag is missing or if the command body is not properly constructed.
- ◆ If some data objects are incorrect, the processing state returned is '6988': this error happens if all the required tags are present but some lengths are different from the ones expected.
- ◆ If the verification of the cryptographic checksum fails, the processing state returned is '6688'.

#### Command with short EF identifier

This command variant enables the IFD to select an EF by means of a short EF identifier and write data from this EF.

TCS\_61 A tachograph card shall support this command variant for all Elementary Files with a specified short EF identifier. These short EF identifiers are specified in chapter 4.

#### TCS\_62 Command Message

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
CLA	1	'00h'	
INS	1	'D6h'	Update Binary
P1	1	'XXh'	Bit 8 is set to 1 Bit 7 and 6 are set to 00 Bit 5 – 1 encode the short EF identifier of the corresponding EF
P2	1	'XXh'	Encodes an offset from 0 to 255 bytes in the EF referenced by P1
Lc	1	'NNh'	Lc Length of data to Update. Number of bytes to be

<i>Byte</i>	<i>Length</i> <i>h</i>	<i>Value</i>	<i>Description</i>
			written.
#6- #(5+NN )	NN	'XX..X Xh'	Data to be written

TCS\_63 **Response Message**

<i>Byte</i>	<i>Length</i> <i>h</i>	<i>Value</i>	<i>Description</i>
SW	2	'XXX Xh'	Status Words (SW1,SW2)

Note: The short EF identifiers used for the generation 2 tachograph application are specified in chapter 4.

If P1 encodes a short EF identifier and the command is successful, the identified EF becomes the currently selected EF (current EF).

- ◆ If the command is successful, the card returns '**9000**'.
- ◆ If the file corresponding with the short EF identifier is not found, the processing state returned is '**6A82**'.
- ◆ If the security conditions of the selected file are not satisfied, the command is interrupted with '**6982**'.
- ◆ If the Offset is not compatible with the size of the EF (Offset > EF size), the processing state returned is '**6B00**'.
- ◆ If the size of the data to be written is not compatible with the size of the EF (Offset + Lc > EF size) the processing state returned is '**6700**'.
- ◆ If an integrity error is detected within the file attributes, the card shall consider the file as corrupted and unrecoverable, the processing state returned is '**6400**' or '**6581**'.
- ◆ If writing is unsuccessful, the processing state returned is '**6581**'.

Command with odd instruction byte

This command variant enables the IFD to write data to an EF with 32768 bytes or more.

TCS\_64 A tachograph card which supports EFs with 32768 bytes or more shall support this command variant for these EFs. A tachograph card may or may not support this command variant for other EFs.

TCS\_65 **Command Message**

<i>Byte</i>	<i>Length</i> <i>h</i>	<i>Value</i>	<i>Description</i>
CLA	1	'00h'	
INS	1	'D7h'	Update Binary
P1	1	'00h'	Current EF
P2	1	'00h'	

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
Lc	1	'NNh'	Lc Length of data in the command data field
#6- #(5+NN )	NN	'XX..X Xh'	Offset data object with tag '54h'    Discretionary data object with tag '53h' that encapsulates the data to be written

The IFD shall encode the offset data object's and the discretionary data object's length with the minimum possible number of octets, i.e. using the length byte '01h' the IFD shall encode an offset / length from 0 to 255 and using the length byte '02h' an offset / length from '256' up to '65535' bytes.

#### TCS\_66 Response Message

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
SW	2	'XXX Xh'	Status Words (SW1,SW2)

- ◆ If the command is successful, the card returns '9000'.
- ◆ If no EF is selected, the processing state returned is '6986'.
- ◆ If the security conditions of the selected file are not satisfied, the command is interrupted with '6982'.
- ◆ If the Offset is not compatible with the size of the EF (Offset > EF size), the processing state returned is '6B00'.
- ◆ If the size of the data to be written is not compatible with the size of the EF (Offset + Lc > EF size) the processing state returned is '6700'.
- ◆ If an integrity error is detected within the file attributes, the card shall consider the file as corrupted and unrecoverable, the processing state returned is '6400' or '6500'.
- ◆ If writing is unsuccessful, the processing state returned is '6581'.

Command with secure messaging (example)

The following example illustrates the usage of secure messaging if the security condition SM-MAC-G2 applies.

#### TCS\_67 Command message

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
CLA	1	'0Ch'	Secure Messaging asked
INS	1	'D7h'	Update Binary
P1	1	'00h'	Current EF
P2	1	'00h'	
Lc	1	'XXh'	Length of the secured data field
#6	1	'B3h'	Tag for plain value data encoded in BER-TLV
#7	L	'NNh'	L <sub>PV</sub> : length of transmitted data.

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
		or '81 NNh'	L is 2 bytes if $L_{PV} > 127$ bytes.
#(7+L)-#(6+L+NN)	N N	'XX..X Xh'	Plain Data encoded in BER-TLV, i.e. offset data object with tag '54h'    Discretionary data object with tag '53h' that encapsulates the data to be written
#(7+L+NN)	1	'8Eh'	T <sub>CC</sub> : Tag for cryptographic checksum
#(8+L+NN)	1	'XXh'	L <sub>CC</sub> : Length of following cryptographic checksum  '08h', '0Ch' or '10h' depending on AES key length for Generation 2 secure messaging (see Appendix 11 Part B)
#(9+L+NN)- #(8+M+L+NN)	M	'XX..X Xh'	Cryptographic checksum
Le	1	'00h'	As specified in ISO/IEC 7816-4

TCS\_68 Response message if the command is successful

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
#1	1	'99h'	T <sub>SW</sub> : Tag for Status Words (to be protected by CC)
#2	1	'02h'	L <sub>SW</sub> : length of returned Status Words
#3- #4	2	'XXX Xh'	Processing Status of the unprotected response APDU
#5	1	'8Eh'	T <sub>CC</sub> : Tag for cryptographic checksum
#6	1	'XXh'	L <sub>CC</sub> : Length of following cryptographic checksum  '08h', '0Ch' or '10h' depending on AES key length for Generation 2 secure messaging (see Appendix 11 Part B)
#7- #(6+ L)	L	'XX..X Xh'	Cryptographic checksum
SW	2	'XXX Xh'	Status Words (SW1,SW2)

## 3.5.4 GET CHALLENGE

This command is compliant with ISO/IEC 7816-4, but has a restricted usage compared to the command defined in the norm.

The GET CHALLENGE command asks the card to issue a challenge in order to use it in a security related procedure in which a cryptogram or some ciphered data are sent to the card.

TCS\_69 The Challenge issued by the card is only valid for the next command, which uses a challenge, sent to the card.

#### TCS\_70 Command Message

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
CLA	1	'00h'	
INS	1	'84h'	INS
P1	1	'00h'	P1
P2	1	'00h'	P2
Le	1	'08h'	Le (Length of Challenge expected).

#### TCS\_71 Response Message

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
#1-#8	8	'XX..XXh'	Challenge
SW	2	'XXXXh'	Status Words (SW1,SW2)

- ◆ If the command is successful, the card returns '9000'.
- ◆ If Le is different from '08h', the processing state is '6700'.
- ◆ If parameters P1-P2 are incorrect, the processing state is '6A86'.

#### 3.5.5 VERIFY

This command is compliant with ISO/IEC 7816-4, but has a restricted usage compared to the command defined in the norm.

Only the workshop card is required to support this command.

Other types of tachograph cards may or may not implement this command, but for these cards no reference CHV is personalized. Therefore these cards cannot perform this command successfully. For other types of tachograph cards than workshop cards the behavior, i.e. the error code returned, is out of the scope of this specification, if this command is sent.

The Verify command initiates the comparison in the card of the CHV (PIN) data sent from the command with the reference CHV stored in the card.

TCS\_72 The PIN entered by the user must be ASCII encoded and right padded with 'FFh' bytes up to a length of 8 bytes by the IFD, see also the data type WorkshopCardPIN in Appendix 1.

TCS\_73 The tachograph applications generation 1 and 2 shall use the same reference CHV.

TCS\_74 The tachograph card shall check whether the command is encoded correctly. If the command is not encoded correctly the card shall not compare the CHV values, not decrement the remaining CHV attempt counter and not reset the security status "PIN\_Verified", but abort the command. A command is encoded correctly, if the CLA, INS, P1, P2, Lc bytes have the specified values, Le is absent, and the command data field has the correct length.

TCS\_75 If the command is successful, the remaining CHV attempt counter is reinitialised. The initial value of the remaining CHV attempt counter is 5. If the command is successful the card shall set the internal security status "PIN\_Verified". The card shall reset this security status, if the card is reset or if the CHV code transmitted in the command does not match the stored reference CHV.

Note: Using the same reference CHV and a global security status prevents that a workshop employee must re-enter the PIN after a selection of another tachograph application DF.

TCS\_76 An unsuccessful comparison is recorded in the card, i.e. the remaining CHV attempts counter shall be decremented by one, in order to limit the number of further attempts of the use of the reference CHV.

TCS\_77 **Command Message**

Byte	Length	Value	Description
CLA	1	'00h'	
INS	1	'20h'	INS
P1	1	'00h'	P1
P2	1	'00h'	P2 (the verified CHV is implicitly known)
Lc	1	'08h'	Length of CHV code transmitted
#6- #13	8	'XX..X Xh'	CHV

TCS\_78 **Response Message**

Byte	Length	Value	Description
SW	2	'XXX Xh'	Status Words (SW1,SW2)

- If the command is successful, the card returns '**9000**'.
- If the reference CHV is not found, the processing state returned is '**6A88**'.
- If the CHV is blocked, (the remaining attempt counter of the CHV is null), the processing state returned is '**6983**'. Once in that state, the CHV can never be successfully presented anymore.
- If the comparison is unsuccessful, the remaining attempt Counter is decreased and the status '**63CX**' is returned (X>0 and X equals the remaining CHV attempts counter).
- If the reference CHV is considered corrupted, the processing state returned is '**6400**' or '**6581**'.
- If Lc is different from '08h', the processing state is '**6700**'.

3.5.6 GET RESPONSE

This command is compliant with ISO/IEC 7816-4.



This command (only necessary and available for T=0 Protocol) is used to transmit prepared data from the card to the interface device (case where a command had included both Lc and Le).

The GET RESPONSE command has to be issued immediately after the command preparing the data, otherwise, the data are lost. After the execution of the GET RESPONSE command (except if the error '61xx' or '6Cxx' occur, see below), the previously prepared data are no longer available.

#### TCS\_79 Command Message

Byte	Length	Value	Description
CLA	1	'00h'	
INS	1	'C0h'	
P1	1	'00h'	
P2	1	'00h'	
Le	1	'XXh'	Number of bytes expected

#### TCS\_80 Response Message

Byte	Length	Value	Description
#1- #X	X	'XX..X Xh'	Data
SW	2	'XXX Xh'	Status Words (SW1,SW2)

- ◆ If the command is successful, the card returns '9000'.
- ◆ If no data have been prepared by the card, the processing state returned is '6900' or '6F00'.
- ◆ If Le exceeds the number of available bytes or if Le is null, the processing state returned is '6Cxx', where xx denotes the exact number of available bytes. In that case, the prepared data are still available for a subsequent GET RESPONSE command.
- ◆ If Le is not null and is smaller than the number of available bytes, the required data are sent normally by the card, and the processing state returned is '61xx', where 'xx' indicates a number of extra bytes still available by a subsequent GET RESPONSE command.
- ◆ If the command is not supported (protocol T=1), the card returns '6D00'.

#### 3.5.7 PSO: VERIFY CERTIFICATE

This command is compliant with ISO/IEC 7816-8, but has a restricted usage compared to the command defined in the norm.

The VERIFY CERTIFICATE command is used by the card to obtain a Public Key from the outside and to check its validity.

#### Generation 1 Command – Response pair

TCS\_81 This command variant is only supported by a generation 1 tachograph application.

TCS\_82 When a VERIFY CERTIFICATE command is successful, the Public Key is stored for a future use in the Security environment. This key shall be explicitly set for the use in security related commands (INTERNAL AUTHENTICATE, EXTERNAL AUTHENTICATE or VERIFY CERTIFICATE) by the MSE command (see § 3.5.11) using its key identifier.

TCS\_83 In any case, the VERIFY CERTIFICATE command uses the public key previously selected by the MSE command to open the certificate. This public key must be the one of a Member State or of Europe.

TCS\_84 **Command Message**

Byte	Length	Value	Description
CLA	1	'00h'	
INS	1	'2Ah'	Perform Security Operation
P1	1	'00h'	P1
P2	1	'AEh'	P2 : non BER-TLV coded data (concatenation of data elements)
Lc	1	'C2h'	Lc : Length of the certificate, 194 bytes
#6- #19 9	194	'XX...X Xh'	Certificate : concatenation of data elements (as described in Appendix 11)

TCS\_85 **Response Message**

Byte	Length	Value	Description
SW	2	'XXXXh'	Status Words (SW1,SW2)

- ◆ If the command is successful, the card returns '9000'.
- ◆ If the certificate verification fails, the processing state returned is '6688'. The verification and unwrapping process of the certificate is described in Appendix 11 for G1 and G2.
- ◆ If no Public Key is present in the Security Environment, '6A88' is returned.
- ◆ If the selected public key (used to unwrap the certificate) is considered corrupted, the processing state returned is '6400' or '6581'.
- ◆ Generation 1 only: If the selected public key (used to unwrap the certificate) has a CHA.LSB (CertificateHolderAuthorisation.equipmentType) different from '00' (i.e. is not the one of a Member State or of Europe), the processing state returned is '6985'.

Generation 2 Command – Response pair

Depending on the curve size ECC certificates may be so long that they cannot be transmitted in a single APDU. In this case command chaining according to ISO/IEC 7816-4 must be applied and the certificate transmitted in two consecutive PSO: Verify Certificate APDUs.

The certificate structure and the domain parameters are defined in Appendix 11.

TCS\_86 The command can be performed in the MF, DF Tachograph and DF Tachograph\_G2, see also TCS\_33.

TCS\_87 **Command Message**

Byte	Length	Value	Description
CLA	1	'X0h'	CLA byte indicating command chaining: '00h' the only or last command of the chain '10h' not the last command of a chain
INS	1	'2Ah'	Perform Security Operation
P1	1	'00h'	
P2	1	'BEh'	Verify self-descriptive certificate
Lc	1	'XXh'	Length of the command data field, see TCS_88 and TCS_89.
#6- #5+ L	L	'XX..X Xh'	DER-TLV encoded data: ECC Certificate Body data object as first data object concatenated with the ECC Certificate Signature data object as second data object or a part of this concatenation. The tag '7F21' and the corresponding length shall not be transmitted.  The order of these data objects is fixed.

TCS\_88 For short length APDUs the following provisions apply: The IFD shall use the minimum number of APDUs required to transmit the command payload and transmit the maximum number of bytes in the first command APDU according to the value of the Information Field Size Card Byte, see TCS\_14. If the IFD behaves differently, the behavior of the card is out of scope.

TCS\_89 For extended length APDUs the following provisions apply: If the certificate does not fit into a single APDU, the card shall support command chaining. The IFD shall use the minimum number of APDUs required to transmit the command payload and transmit the maximum number of bytes in the first command APDU. If the IFD behaves differently, the behavior of the card is out of scope.

Note: According to Appendix 11 the card stores the certificate or the relevant contents of the certificate and updates its currentAuthenticatedTime.

The response message structure and status words are as defined in TCS\_85.

TCS\_90 In addition to the error codes listed in TCS\_85, the card may return the following error codes:

- ◆ If the selected public key (used to unwrap the certificate) has a CHA.LSB (CertificateHolderAuthorisation.equipmentType) that is not suitable for the certificate verification according to Appendix 11, the processing state returned is '**6985**'.
- ◆ If the currentAuthenticatedTime of the card is later than the Certificate Expiration Date, the processing state returned is '**6985**'.
- ◆ If the last command of the chain is expected, the card returns '**6883**'.
- ◆ If incorrect parameters are sent in the command data field, the card returns '**6A80**' (also used in case the data objects are not sent in the specified order).

### 3.5.8 INTERNAL AUTHENTICATE

This command is compliant with ISO/IEC 7816-4.

TCS\_91 All tachograph cards shall support this command in the DF Tachograph generation 1. The command may or may not be accessible in the MF and / or the DF Tachograph\_G2. If so, the command shall terminate with a suitable error code as the private key of the card (Card.SK) for the generation 1 authentication protocol is only accessible in the DF\_Tachograph generation 1.

Using the INTERNAL AUTHENTICATE command, the IFD can authenticate the card. The authentication process is described in Appendix 11. It includes the following statements:

TCS\_92 The INTERNAL AUTHENTICATE command uses the card Private Key (implicitly selected) to sign authentication data including K1 (first element for session key agreement) and RND1, and uses the Public Key currently selected (through the last MSE command) to encrypt the signature and form the authentication token (more details in Appendix 11).

TCS\_93 **Command Message**

<i>Byte</i>	<i>Length</i> <i>h</i>	<i>Value</i>	<i>Description</i>
CLA	1	'00h'	CLA
INS	1	'88h'	INS
P1	1	'00h'	P1
P2	1	'00h'	P2
Lc	1	'10h'	Length of data sent to the card
#6 - #13	8	'XX..XXh'	Challenge used to authenticate the card
#14 -#21	8	'XX..XXh'	VU.CHR (see Appendix 11)
Le	1	'80h'	Length of the data expected from the card

TCS\_94 **Response Message**

<i>Byte</i>	<i>Length</i> <i>h</i>	<i>Value</i>	<i>Description</i>
#1-#128	128	'XX..XXh'	Card authentication token (see Appendix 11)
SW	2	'XXXXh'	Status Words (SW1,SW2)

- ◆ If the command is successful, the card returns '9000'.
- ◆ If no Public Key is present in the Security Environment, the processing state returned is '6A88'.
- ◆ If no Private Key is present in the Security Environment, the processing state returned is '6A88'.
- ◆ If VU.CHR does not match the current public key identifier, the processing state returned is '6A88'.
- ◆ If the selected private key is considered corrupted, the processing state returned is '6400' or '6581'.

TCS\_95 If the INTERNAL AUTHENTICATE command is successful, the current session key, if existing, is erased and no longer available. In order to have a new session key available, the EXTERNAL AUTHENTICATE command for the generation 1 authentication mechanism must be successfully performed.

### 3.5.9 EXTERNAL AUTHENTICATE

This command is compliant with ISO/IEC 7816-4.

Using the EXTERNAL AUTHENTICATE command, the card can authenticate the IFD. The authentication process is described in Appendix 11 for Tachograph G1 and G2 (VU authentication).

TCS\_96 The command variant for the generation 1 mutual authentication mechanism is only supported by a generation 1 tachograph application.

TCS\_97 The command variant for the second generation VU-card mutual authentication can be performed in the MF, DF Tachograph and DF Tachograph\_G2, see also TCS\_34.

TCS\_98 **Command Message**

<i>Byte</i>	<i>Length</i> <i>h</i>	<i>Value</i>	<i>Description</i>
CLA	1	'00h'	CLA
INS	1	'82h'	INS
P1	1	'00h'	Keys and algorithms implicitly known
P2	1	'00h'	
Lc	1	'XXh'	Lc (Length of the data sent to the card )
#6-#(5+L)	L	'XX..XXh'	Generation 1 authentication: Cryptogram (see Appendix 11 Part A) Generation 2 authentication: Signature generated by the IFD (see Appendix 11 Part B)

TCS\_99 **Response Message**

<i>Byte</i>	<i>Length</i> <i>h</i>	<i>Value</i>	<i>Description</i>
SW	2	'XXXXh'	Status Words (SW1,SW2)

- If the command is successful, the card returns '**9000**'.
- If the CHA of the currently set public key is not the concatenation of the Tachograph application AID and of a VU equipment Type, the processing state returned is '**6F00**'.
- If the command is not immediately preceded with a GET CHALLENGE command, the processing state returned is '**6985**'.

The Generation 1 Tachograph application may return the following additional error codes:

- If no Public Key is present in the Security Environment, '**6A88**' is returned.
- If no Private Key is present in the Security Environment, the processing state returned is '**6A88**'.
- If the verification of the cryptogram is wrong, the processing state returned is '**6688**'.
- If the selected private key is considered corrupted, the processing state returned is '**6400**' or '**6581**'.

The command variant for the Generation 2 authentication may return the following additional error code:

- If signature verification failed, the card returns '**6300**'.

## 3.5.10 GENERAL AUTHENTICATE

This command is used for the generation 2 chip authentication protocol specified in Appendix 11 Part B and is compliant with ISO/IEC 7816-4.

TCS\_100 The command can be performed in the MF, DF Tachograph and DF Tachograph\_G2, see also TCS\_34.

TCS\_101 **Command Message**

Byte	Length h	Value	Description
CLA	1	'00h'	
INS	1	'86h'	
P1	1	'00h'	Keys and protocol implicitly known
P2	1	'00h'	
Lc	1	'NNh'	Lc : length of subsequent data field
#6-#(5+L)	L	'7Ch' + L <sub>7C</sub> + '80h' + L <sub>80</sub> + 'XX..XXh'	DER-TLV encoded ephemeral public key value (see Appendix 11) The VU shall send the data objects in this order.

TCS\_102 **Response Message**

Byte	Length h	Value	Description
#1-#L	L	'7Ch' + L <sub>7C</sub> + '81h' + '08h' + 'XX..XXh' + '82h' + L <sub>82</sub> + 'XX..XXh'	DER-TLV encoded Dynamic Authentication Data: nonce and authentication token (see Appendix 11)
SW	2	'XXXXh'	Status Words (SW1,SW2)

- ◆ If the command is successful, the card returns '9000'.
- ◆ The card returns '6A80' to indicate incorrect parameters in data field.
- ◆ The card returns '6982' if the External Authenticate command has not been performed successfully

The response Dynamic Authentication Data object '7Ch'

- must be present if the operation is successful, i.e. the Status Words are '9000',
- must be absent in case of an execution error or checking error, i.e. if the Status Words are in the range '6400' – '6FFF', and
- may be absent in case of a warning, i.e. if the Status Words are in the range '6200' – '63FF'.

## 3.5.11 MANAGE SECURITY ENVIRONMENT

This command is used to set a public key for authentication purpose.

#### Generation 1 Command – Response pair

This command is compliant with ISO/IEC 7816-4. The use of this command is restricted regarding the related standard.

TCS\_103 This command is only supported by a generation 1 tachograph application.

TCS\_104 The key referenced in the MSE data field remains the current public key until the next correct MSE command, a DF is selected or the card is reset.

TCS\_105 If the key referenced is not (already) present into the card, the security environment remains unchanged.

#### TCS\_106 **Command Message**

<i>Byte</i>	<i>Length</i> <i>h</i>	<i>Value</i>	<i>Description</i>
CLA	1	'00h'	CLA
INS	1	'22h'	INS
P1	1	'C1h'	P1 : referenced key valid for all cryptographic operations
P2	1	'B6h'	P2 (referenced data concerning Digital Signature)
Lc	1	'0Ah'	Lc : length of subsequent data field
#6	1	'83h'	Tag for referencing a public key in asymmetric cases
#7	1	'08h'	Length of the key reference (key identifier)
#8- #15	8	'XX..X Xh'	Key identifier as specified in Appendix 11

#### TCS\_107 **Response Message**

<i>Byte</i>	<i>Length</i> <i>h</i>	<i>Value</i>	<i>Description</i>
SW	2	'XXX Xh'	Status Words (SW1,SW2)

- If the command is successful, the card returns '9000'.
- If the referenced key is not present into the card, the processing state returned is '6A88'.
- If some expected data objects are missing in the secure messaging format, the processing state '6987' is returned. This can happen if the tag '83h' is missing.
- If some data objects are incorrect, the processing state returned is '6988'. This can happen if the length of the key identifier is not '08h'.
- If the selected key is considered corrupted, the processing state returned is '6400' or '6581'.

## Generation 2 Command – Response pairs

For the Generation 2 authentication the tachograph card supports the following MSE: Set command versions which are compliant with ISO/IEC 7816-4. These command versions are not supported for the Generation 1 authentication.

## 3.5.12 MSE:SET AT for Chip Authentication

The following MSE:SET AT command is used to select the parameters for the Chip Authentication that is performed by a subsequent General Authenticate command.

TCS\_108 The command can be performed in the MF, DF Tachograph and DF Tachograph\_G2, see also TCS\_34.

## TCS\_109 MSE:SET AT Command Message for Chip Authentication

<i>Byte</i>	<i>Length h</i>	<i>Value</i>	<i>Description</i>
CLA	1	'00h'	
INS	1	'22h'	
P1	1	'41h'	Set for internal authentication
P2	1	'A4h'	Authentication
Lc	1	'NNh'	Lc : length of subsequent data field
#6- #(5+L)	L	'80h' + '0Ah' + 'XX..XXh'	DER-TLV encoded cryptographic mechanism reference: Object Identifier of Chip Authentication (value only, Tag '06h' is omitted).  See Appendix 1 for the values of object identifiers; the byte notation shall be used. See Appendix 11 for guidance on how to select one of these object identifiers.

## MSE:SET AT for VU Authentication

The following MSE:SET AT command is used to select the parameters and keys for the VU Authentication that is performed by a subsequent External Authenticate command.

TCS\_110 The command can be performed in the MF, DF Tachograph and DF Tachograph\_G2, see also TCS\_34.

## TCS\_111 MSE:SET AT Command Message for VU Authentication

<i>Byte</i>	<i>Length h</i>	<i>Value</i>	<i>Description</i>
CLA	1	'00h'	
INS	1	'22h'	
P1	1	'81h'	Set for external authentication
P2	1	'A4h'	Authentication
Lc	1	'NNh'	Lc : length of subsequent data field



<i>Byte</i>	<i>Length</i> <i>h</i>	<i>Value</i>	<i>Description</i>
#6- #(5+L)	L	'80h' + '0Ah' + 'XX..XXh'	DER-TLV encoded cryptographic mechanism reference: Object Identifier of VU Authentication (value only, Tag '06h' is omitted).  See Appendix 1 for the values of object identifiers; the byte notation shall be used. See Appendix 11 for guidance on how to select one of these object identifiers.
		'83h' + '08h' + 'XX..XXh'	DER-TLV encoded reference of the VU public key by the Certificate Holder Reference mentioned in its certificate.
		'91h' + L <sub>91</sub> + 'XX..XXh'	DER-TLV encoded compressed representation of the ephemeral public key of the VU that will be used during Chip Authentication (see Appendix 11)

### 3.5.13 MSE:SET DST

The following MSE:SET DST command is used to set a public key either

- ◆ for the verification of a signature that is provided in a subsequent PSO: Verify Digital Signature command or
- ◆ for the signature verification of a certificate that is provided in a subsequent PSO: Verify Certificate command

TCS\_112 The command can be performed in the MF, DF Tachograph and DF Tachograph\_G2, see also TCS\_33.

### TCS\_113 MSE:SET DST Command Message

<i>Byte</i>	<i>Length</i> <i>h</i>	<i>Value</i>	<i>Description</i>
CLA	1	'00h'	
INS	1	'22h'	
P1	1	'81h'	Set for verification
P2	1	'B6h'	Digital Signature
Lc	1	'NNh'	Lc : length of subsequent data field
#6-#(5+L)	L	'83h' + '08h' + 'XX...XXh'	DER-TLV encoded reference of a public key, i.e. the Certificate Holder Reference in the certificate of the public key (see Appendix 11)

For all command versions the response message structure and status words are given by:

### TCS\_114 Response Message

<i>Byte</i>	<i>Length</i> <i>h</i>	<i>Value</i>	<i>Description</i>
SW	2	'XXXXh'	Status Words (SW1,SW2)

- ◆ If the command is successful, the card returns '9000'. The protocol has been selected and initialised.
- ◆ '6A80' indicates incorrect parameters in the command data field.
- ◆ '6A88' indicates that referenced data (i.e. a referenced key) is not available.

#### 3.5.14 PSO: HASH

This command is used to transfer to the card the result of a hash calculation on some data. This command is used for the verification of digital signatures. The hash value is stored temporarily for the subsequent command PSO: Verify Digital Signature

This command is compliant with ISO/IEC 7816-8. The use of this command is restricted regarding the related standard.

Only the control card is required to support this command in the DF Tachograph and DF Tachograph\_G2.

Other types of tachograph cards may or may not implement this command. The command may or may not be accessible in the MF.

The control card application generation 1 supports only SHA-1.

TCS\_115 The temporarily stored hash value shall be deleted if a new hash value is computed by means of the PSO: HASH command, if a DF is selected, and if the tachograph card is reset.

#### TCS\_116 Command Message

<i>Byte</i>	<i>Length</i> <i>h</i>	<i>Value</i>	<i>Description</i>
CLA	1	'00h'	CLA
INS	1	'2Ah'	Perform Security Operation
P1	1	'90h'	Return Hash code
P2	1	'A0h'	Tag : data field contains DOs relevant for hashing
Lc	1	'XXh'	Length Lc of the subsequent data field
#6	1	'90h'	Tag for the hash code
#7	1	'XXh'	Length L of the hash code: '14h' in Generation 1 application (see Appendix 11 Part A) '20h', '30h' or '40h' in Generation 2 application (see Appendix 11 Part B)
#8- #(7+ L)	L	'XX..X Xh'	Hash code

#### TCS\_117 Response Message

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
SW	2	'XXXXh'	Status Words (SW1,SW2)

- If the command is successful, the card returns '**9000**'.
- If some expected data objects (as specified above) are missing, the processing state '**6987**' is returned. This can happen if one of the tag '90h' is missing.
- If some data objects are incorrect, the processing state returned is '**6988**'. This error happens if the required tag is present but with a length different from '14h' for SHA-1, '20h' for SHA-256, '30h' for SHA-384, '40h' for SHA-512 (Generation 2 application).

### 3.5.15 PERFORM HASH of FILE

This command is not compliant with ISO/IEC 7816-8. Thus the CLA byte of this command indicates that there is a proprietary use of the PERFORM SECURITY OPERATION / HASH.

Only the driver card and the workshop card are required to support this command in the DF Tachograph and DF Tachograph\_G2.

Other types of tachograph cards may or may not implement this command. If a company or control card implements this command, the command shall be implemented as specified in this chapter.

The command may or may not be accessible in the MF. If so, the command shall be implemented as specified in this chapter, i.e. shall not allow the calculation of a hash value, but terminate with a suitable error code.

TCS\_118 The PERFORM HASH of FILE command is used to hash the data area of the currently selected transparent EF.

TCS\_119 A tachograph card shall support this command only for the EFs that are listed in chapter 4 under the DF\_Tachograph and DF\_Tachograph\_G2 with the following exception. A tachograph card shall not support the command for the EF Sensor\_Installation\_Data of DF Tachograph\_G2..

TCS\_120 The result of the hash operation is stored temporarily in the card. It can then be used to get a digital signature of the file, using the PSO: COMPUTE DIGITAL SIGNATURE command.

TCS\_121 The temporarily stored hash of file value shall be deleted if a new hash of file value is computed by means of the PSO: Hash of File command, if a DF is selected, and if the tachograph card is reset.

TCS\_122 The Tachograph Generation 1 application shall support SHA-1.

TCS\_123 The Tachograph Generation 2 application shall support SHA-1 and SHA-2 (256, 384 and 512 bits).

TCS\_124 **Command Message**

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
CLA	1	'80h'	CLA
INS	1	'2Ah'	Perform Security Operation
P1	1	'90h'	Tag: Hash
P2	1	'XXh'	P2 : Indicates the algorithm to be used for hashing of the data of the currently selected transparent file:
			'00h' for SHA-1

<i>Byte</i>	<i>Length</i> <i>h</i>	<i>Value</i>	<i>Description</i>
			'01h' for SHA-256
			'02h' for SHA-384
			'03h' for SHA-512

TCS\_125 **Response Message**

<i>Byte</i>	<i>Length</i> <i>h</i>	<i>Value</i>	<i>Description</i>
SW	2	'XX XXh'	Status Words (SW1,SW2)

- If the command is successful, the card returns **'9000'**.
- If the current EF does not allow this command (EF Sensor\_Installation\_Data in DF Tachograph\_G2), the processing state **'6985'** is returned.
- If the selected EF is considered corrupted (file attributes or stored data integrity errors), the processing state returned is **'6400'** or **'6581'**.
- If the selected file is not a transparent file or if there is no current EF, the processing state returned is **'6986'**.

3.5.16 PSO: COMPUTE DIGITAL SIGNATURE

This command is used to compute the digital signature of previously computed hash code (see PERFORM HASH of FILE, §3.5.15).

Only the driver card and the workshop card are required to support this command in the DF Tachograph and DF Tachograph\_G2.

Other types of tachograph cards may or may not implement this command, but shall not have a signature key. Therefore these cards cannot perform the command successfully, but terminate with a suitable error code.

The command may or may not be accessible in the MF. If so, the command shall terminate with a suitable error code.

This command is compliant with ISO/IEC 7816-8. The use of this command is restricted regarding the related standard.

TCS\_126 This command shall not compute a digital signature of previously computed hash code with the PSO: HASH command.

TCS\_127 The card private key is used to compute the digital signature and is implicitly known by the card.

TCS\_128 The Generation 1 tachograph application performs a digital signature using a padding method compliant with PKCS1 (see Appendix 11 for details).

TCS\_129 The Generation 2 tachograph application computes an elliptic curve based digital signature (see Appendix 11 for details).

TCS\_130 **Command Message**

<i>Byte</i>	<i>Length</i> <i>h</i>	<i>Value</i>	<i>Description</i>
CLA	1	'00h'	CLA

<i>Byte</i>	<i>Length</i> <i>h</i>	<i>Value</i>	<i>Description</i>
INS	1	'2Ah'	Perform Security Operation
P1	1	'9Eh'	Digital signature to be returned
P2	1	'9Ah'	Tag: data field contains data to be signed. As no data field is included, the data are supposed to be already present in the card (hash of file)
Le	1	'NNh'	Length of the expected signature

TCS\_131 **Response Message**

<i>Byte</i>	<i>Length</i> <i>h</i>	<i>Value</i>	<i>Description</i>
#1- #L	L	'XX..X Xh'	Signature of the previously computed hash
SW	2	'XXX Xh'	Status Words (SW1,SW2)

- If the command is successful, the card returns '**9000**'.
- If the implicitly selected private key is considered as corrupted, the processing state returned is '**6400**' or '**6581**'.
- If the hash which was computed in a previous Perform Hash of File command is not available, the processing state returned is '**6985**'.

## 3.5.17 PSO: VERIFY DIGITAL SIGNATURE

This command is used to verify the digital signature, provided as an input, whose hash is known to the card. The signature algorithm is implicitly known by the card.

This command is compliant with ISO/IEC 7816-8. The use of this command is restricted regarding the related standard.

Only the control card is required to support this command in the DF Tachograph and DF Tachograph\_G2.

Other types of tachograph cards may or may not implement this command. The command may or may not be accessible in the MF.

TCS\_132 The VERIFY DIGITAL SIGNATURE command always uses the public key selected by the previous Manage Security Environment MSE: Set DST command and the previous hash code entered by a PSO: HASH command.

TCS\_133 **Command Message**

<i>Byte</i>	<i>Length</i> <i>h</i>	<i>Value</i>	<i>Description</i>
CLA	1	'00h'	CLA
INS	1	'2Ah'	Perform Security Operation
P1	1	'00h'	
P2	1	'A8h'	Tag : data field contains DOs relevant for

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
			verification
Lc	1	'83h'	Length Lc of the subsequent data field
6	1	'9Eh'	Tag for Digital Signature
#7-#8	2	'81 XXh'	Length of digital signature: 128 bytes coded in accordance with Appendix 11 Part A for Tachograph Generation 1 application Depending on the selected curve for Tachograph Generation 2 application (see Appendix 11 Part B)
#9- #(8+L)	L	'XX..XXh',	Digital signature content

TCS\_134 **Response Message**

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
SW	2	'XXXXh'	Status Words (SW1,SW2)

- ◆ If the command is successful, the card returns '**9000**'.
- ◆ If the verification of the signature fails, the processing state returned is '**6688**'. The verification process is described in Appendix 11.
- ◆ If no public key is selected, the processing state returned is '**6A88**'.
- ◆ If some expected data objects (as specified above) are missing, the processing state '**6987**' is returned. This can happen if one of the required tag is missing.
- ◆ If no hash code is available to process the command (as a result of a previous PSO: Hash command), the processing state returned is '**6985**'.
- ◆ If some data objects are incorrect, the processing state returned is '**6988**'. This can happen if one of the required data objects length is incorrect.
- ◆ If the selected public key is considered corrupted, the processing state returned is '**6400**' or '**6581**'.

## 3.5.18 PROCESS DSRC MESSAGE

This command is used to verify the integrity and authenticity of the DSRC message and to decipher the data communicated from a VU to a control authority or a workshop over the DSRC link. The card derives the encryption key and the MAC key used to secure the DSRC message as described in Appendix 11 Part B chapter 13.

Only the control card and the workshop card are required to support this command in the DF Tachograph\_G2.

Other types of tachograph cards may or may not implement this command, but shall not have a DSRC master key. Therefore these cards cannot perform the command successfully, but terminate with a suitable error code.

The command may or may not be accessible in the MF and / or the DF Tachograph. If so, the command shall terminate with a suitable error code.

TCS\_135 The DSRC master key is accessible only in the DF Tachograph\_G2, i.e. the control and workshop card shall support a successful execution of the command only in the DF Tachograph\_G2.

TCS\_136 The command shall only decrypt the DSRC data and verify the cryptographic checksum, but not interpret the input data.

TCS\_137 The order of the data objects in the command data field is fixed by this specification.

TCS\_138 **Command Message**

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
CLA	1	'80h'	Proprietary CLA
INS	1	'2Ah'	Perform Security Operation
P1	1	'80h'	Response data: plain value
P2	1	'B0h'	Command data: plain value encoded in BER-TLV and including SM DOs
Lc	1	'NNh'	Length Lc of the subsequent data field
#6- #(5+L)	L	'87h' + L <sub>87</sub> + 'XX..XXh' ,	<p>DER-TLV encoded padding-content indicator byte followed by encrypted tachograph payload. For the padding-content indicator byte the value '00h' ('no further indication' according to ISO/IEC 7816-4:2013 Table 52) shall be used. For the encryption mechanism see Appendix 11, Part B chapter 13.</p> <p>Allowed values for the length L<sub>87</sub> are the multiples of the AES block length plus 1 for the padding-content indicator byte, i.e. from 17 bytes up to and including 193 bytes.</p> <p>Note: See ISO/IEC 7816-4:2013 Table 49 for the SM data object with tag '87h'.</p>
		'81h' + '10h'	<p>DER-TLV encoded Control Reference Template for Confidentiality nesting the concatenation of the following data elements (see Appendix 1 DSRCSecurityData and Appendix 11 Part B chapter 13):</p> <ul style="list-style-type: none"> <li>• 4 byte time stamp</li> <li>• 3 byte counter</li> <li>• 8 byte VU serial number</li> <li>• 1 byte DSRC master key version</li> </ul> <p>Note: See ISO/IEC 7816-4:2013 Table 49 for the SM data object with tag '81h'.</p>
		'8Eh' + L <sub>8E</sub> + 'XX..XXh' ,	<p>DER-TLV encoded MAC over the DSRC message. For the MAC algorithm and calculation see Appendix 11, Part B chapter 13.</p> <p>Note: See ISO/IEC 7816-4:2013 Table 49 for the</p>

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
			SM data object with tag '8Eh'.

TCS\_139 **Response Message**

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
#1-#L	L	'XX..XXh'	Absent (in case of an error) or deciphered data (padding removed)
SW	2	'XXXXh'	Status Words (SW1,SW2)

- If the command is successful, the card returns '9000'.
- '6A80' indicates incorrect parameters in the command data field (also used in case the data objects are not sent in the specified order).
- '6A88' indicates that referenced data is not available, i.e. the referenced DSRC master key is not available.
- '6900' indicates that the verification of the cryptographic checksum or the decryption of the data failed

4. Tachograph cards structure

This paragraph specifies the file structures of the Tachograph cards for storage of accessible data.

It does not specify card manufacturer dependent internal structures, such as e.g. file headers, nor storage and handling of data elements needed for internal use only such as EuropeanPublicKey, CardPrivateKey, TDesSessionKey or WorkshopCardPin.

TCS\_140 A generation 2 tachograph card shall host the Master File MF and a generation 1 and a generation 2 tachograph application of the same type (e.g. driver card applications).

TCS\_141 A tachograph card shall support at least the minimum number of records specified for the corresponding applications and shall not support more records than the maximum number of records specified for the corresponding applications.

The maximum and minimum numbers of records are specified in this chapter for the different applications.

For the security conditions used in the access rules throughout this chapter please refer to chapter 3.3. In general the access mode "read" denotes the READ BINARY command with even and if supported odd INS byte with the exception of the EF Sensor\_Installation\_Data on the workshop card, see TCS\_156 and TCS\_160. The access mode "update" denotes the Update Binary command with even and if supported odd INS byte and the access mode "select" the SELECT command.

4.1. Master File MF

TCS\_142 After its personalisation, the master file MF shall have the following permanent file structure and file access rules:

Note: .The short EF identifier SFID is given as decimal number, e.g. the value 30 corresponds to 11110 in binary.

<b>File</b>	<i>Access rules</i>			
	<b>File ID</b>	<b>SFI D</b>	<b>Read / Select</b>	<b>Update</b>
MF	'3F00h'			



					<i>Access rules</i>	
E	ICC	'0002			ALW	NEV
E	IC	'0005			ALW	NEV
E	DIR	'2F00	30		ALW	NEV
E	ATR/INFO (conditional)	'2F01	29		ALW	NEV
E	Extended_Length	'0006	28		ALW	NEV
D	Tachograph	'0500			SC1	
D	Tachograph_G2				SC1	

The following abbreviation for the security condition is used in this table:

**SC1**ALW OR SM-MAC-G2

TCS\_143 All EF structures shall be transparent.

TCS\_144 The Master File MF shall have the following data structure:

<b>File / Data element</b>	<i>No of Records</i>	<i>Size (bytes)</i>		<i>Default Values</i>
		<b>Min</b>	<b>Max</b>	
MF		63	184	
E ICC		25	25	
CardIccIdentification		25	25	
clockStop		1	1	{00}
cardExtendedSerialNumber		8	8	{00..00}
cardApprovalNumber		8	8	{20..20}
cardPersonaliserID		1	1	{00}
embedderIcAssemblerId		5	5	{00..00}
icIdentifier		2	2	{00 00}
E IC		8	8	
CardChipIdentification		8	8	
icSerialNumber		4	4	{00..00}
icManufacturingReferences		4	4	{00..00}
E DIR		20	20	
See TCS_145		20	20	{00..00}
E ATR/INFO		7	128	
See TCS_146		7	128	{00..00}
E EXTENDED_LENGTH		3	3	
See TCS_147		3	3	{00..00}
D Tachograph				
D Tachograph_G2				
F				

TCS\_145 The elementary file EF DIR shall contain the following application related data objects: '61 08 4F 06 FF 54 41 43 48 4F 61 08 4F 06 FF 53 4D 52 44 54'

TCS\_146 The elementary file EF ATR/INFO shall be present if the tachograph card indicates in its ATR that it supports extended length fields. In this case the EF ATR/INFO shall contain the extended length information data object (DO'7F66') as specified in ISO/IEC 7816-4:2013 clause 12.7.1.

TCS\_147 The elementary file EF Extended\_Length shall be present if the tachograph card indicates in its ATR that it supports extended length fields. In this case the EF shall contain the following data object: '02 01 xx' where the value 'xx' indicates whether extended length fields are supported for the T = 1 and / or T = 0 protocol.  
The value '01' indicates extended length field support for the T = 1 protocol.  
The value '10' indicates extended length field support for the T = 0 protocol.  
The value '11' indicates extended length field support for the T = 1 and the T = 0 protocol.

## 4.2. Driver card applications

### 4.2.1 Driver card application generation 1

TCS\_148 After its personalisation, the driver card application generation 1 shall have the following permanent file structure and file access rules:

File	File ID	Access rules		
		Read	Select	Update
D Tachograph	'0500h		SC1	
E Application_Identification	'0501h	SC	SC1	NEV
E Card_Certificate	'C100h	SC	SC1	NEV
E CA_Certificate	'C108h	SC	SC1	NEV
E Identification	'0520h	SC	SC1	NEV
E Card_Download	'050Eh	SC	SC1	SC1
E Driving_Licence_Info	'0521h	SC	SC1	NEV
E Events_Data	'0502h	SC	SC1	SC3
E Faults_Data	'0503h	SC	SC1	SC3
E Driver_Activity_Data	'0504h	SC	SC1	SC3
E Vehicles_Used	'0505h	SC	SC1	SC3
E Places	'0506h	SC	SC1	SC3
E Current_Usage	'0507h	SC	SC1	SC3
E Control_Activity_Data	'0508h	SC	SC1	SC3
E Specific_Conditions	'0522h	SC	SC1	SC3

The following abbreviations for the security conditions are used in this table:

**SC1**ALW OR SM-MAC-G2

**SC2**ALW OR SM-MAC-G1 OR SM-MAC-G2

**SC3**SM-MAC-G1 OR SM-MAC-G2

TCS\_149 All EF structures shall be transparent.

TCS\_150 The driver card application generation 1 shall have the following data structure:

File / Data element	No of Records	Size (bytes)		Default Values
		Min	Max	
D Tachograph		11378	24926	
EF Application_Identification		10	10	
DriverCardApplicationIdentification		10	10	
typeOfTachographCardId		1	1	{00}
cardStructureVersion		2	2	{00 00}
noOfEventsPerType		1	1	{00}
noOfFaultsPerType		1	1	{00}
activityStructureLength		2	2	{00 00}
noOfCardVehicleRecords		2	2	{00 00}
noOfCardPlaceRecords		1	1	{00}
EF Card_Certificate		194	194	
CardCertificate		194	194	{00..00}
EF CA_Certificate		194	194	
MemberStateCertificate		194	194	{00..00}
E Identification		143	143	
CardIdentification		65	65	
cardIssuingMemberState		1	1	{00}
cardNumber		16	16	{20..20}
cardIssuingAuthorityName		36	36	{20..20}
cardIssueDate		4	4	{00..00}
cardValidityBegin		4	4	{00..00}
cardExpiryDate		4	4	{00..00}
DriverCardHolderIdentification		78	78	
cardHolderName		72	72	
holderSurname		36	36	{00,
holderFirstNames		36	36	{00,
cardHolderBirthDate		4	4	{00..00}
cardHolderPreferredLanguage		2	2	{20 20}
EF Card_Download		4	4	
LastCardDownload		4	4	
EF Driving_Licence_Info		53	53	
CardDrivingLicenceInformation		53	53	
drivingLicenceIssuingAuthority		36	36	{00,
drivingLicenceIssuingNation		1	1	{00}
drivingLicenceNumber		16	16	{20..20}
EF Events_Data		864	1728	
CardEventData		864	1728	
cardEventRecords	6	144	288	
CardEventRecord	n <sub>1</sub>	24	24	
eventType		1	1	{00}
eventBeginTime		4	4	{00..00}
eventEndTime		4	4	{00..00}
eventVehicleRegistration				
vehicleRegistrationNation		1	1	{00}

File / Data element	No of Records	Size (bytes)		Default Values
		Min	Max	
EF Faults_Data		576	1152	00 00
CardFaultData		576	1152	
cardFaultRecords	2	288	576	
CardFaultRecord	n <sub>2</sub>	24	24	
faultType		1	1	{00}
faultBeginTime		4	4	{00..00}
faultEndTime		4	4	{00..00}
faultVehicleRegistration				
vehicleRegistrationNation		1	1	{00}
vehicleRegistrationNumber		14	14	{00,
EF Driver_Activity_Data		5548	13780	
Driver Activitv Data				
CardDriverActivity		5548	13780	
activityPointerOldestDayRecord		2	2	{00 00}
activityPointerNewestRecord		2	2	{00 00}
activityDailyRecords	n <sub>6</sub>	5544	13776	{00..00}
EF Vehicles_Used		2606	6202	
CardVehiclesUsed		2606	6202	
vehiclePointerNewestRecord		2	2	{00 00}
cardVehicleRecords		2604	6200	
CardVehicleRecord	n <sub>3</sub>	31	31	
vehicleOdometerBegin		3	3	{00..00}
vehicleOdometerEnd		3	3	{00..00}
vehicleFirstUse		4	4	{00..00}
vehicleLastUse		4	4	{00..00}
vehicleRegistration				
vehicleRegistrationNation		1	1	{00}
vehicleRegistrationNumber		14	14	{00,
vuDataBlockCounter		2	2	{00 00}
EF Places		841	1121	
CardPlaceDailyWorkPeriod		841	1121	
placePointerNewestRecord		1	1	{00}
placeRecords		840	1120	
PlaceRecord	n <sub>4</sub>	10	10	
entryTime		4	4	{00..00}
entryTypeDailyWorkPeriod		1	1	{00}
dailyWorkPeriodCountry		1	1	{00}
dailyWorkPeriodRegion		1	1	{00}
vehicleOdometerValue		3	3	{00..00}
E Current_Usage		19	19	
CardCurrentUse		19	19	
sessionOpenTime		4	4	{00..00}
sessionOpenVehicle				{00}

File / Data element	No of Records	Size (bytes)		Default Values
		Min	Max	
vehicleRegistrationNumber		14	14	{00,
EF Control_Activity_Data		46	46	
CardControlActivityDataRecord		46	46	
controlType		1	1	{00}
controlTime		4	4	{00..00}
controlCardNumber				
cardType		1	1	{00}
cardIssuingMemberState		1	1	{00}
cardNumber		16	16	{20..20}
controlVehicleRegistration				
vehicleRegistrationNation		1	1	{00}
vehicleRegistrationNumber		14	14	{00,
controlDownloadPeriodBegin		4	4	{00..00}
controlDownloadPeriodEnd		4	4	{00..00}
EF Specific_Conditions		280	280	
SpecificConditionRecord	56	5	5	
entryTime		4	4	{00..00}
SpecificConditionType		1	1	{00}

TCS\_151 The following values, used to provide sizes in the table above, are the minimum and maximum record number values the driver card data structure must use for a generation 1 application:

	Min	Max
n NoOfEventsPerType	6	12
1		
n NoOfFaultsPerType	12	24
2		
n NoOfCardVehicleRecords	84	200
3		
n NoOfCardPlaceRecords	84	112
4		
n CardActivityLengthRange	5544 bytes	13776 Bytes
6	(28 days * 93 activity changes)	(28 days * 240 activity changes)

#### 4.2.2 Driver card application generation 2

TCS\_152 After its personalisation, the driver card application generation 2 shall have the following permanent file structure and file access rules.

Note: .The short EF identifier SFID is given as decimal number, e.g. the value 30 corresponds to 11110 in binary.

					<i>Access rules</i>
<b>File</b>	<b>File ID</b>	<b>SFI D</b>	<b>Read / Select</b>	<b>Update</b>	
D Tachograph_G2			SC1		
E Application_Identification	'0501	1	SC1	NEV	
E CardMA_Certificate	'C10	2	SC1	NEV	
E CardSignCertificate	'C10	3	SC1	NEV	
E CA_Certificate	'C10	4	SC1	NEV	
E Link_Certificate	'C10	5	SC1	NEV	
E Identification	'0520	6	SC1	NEV	
E Card_Download	'050E	7	SC1	SC1	
E Driving_Licence_Info	'0521	10	SC1	NEV	
E Events_Data	'0502	12	SC1	SM-	
E Faults_Data	'0503	13	SC1	SM-	
E Driver_Activity_Data	'0504	14	SC1	SM-	
E Vehicles_Used	'0505	15	SC1	SM-	
E Places	'0506	16	SC1	SM-	
E Current_Usage	'0507	17	SC1	SM-	
E Control_Activity_Data	'0508	18	SC1	SM-	
E Specific_Conditions	'0522	19	SC1	SM-	
E VehicleUnits_Used	'0523	20	SC1	SM-	
E GNSS_Places	'0524	21	SC1	SM-	

The following abbreviation for the security condition is used in this table:

**SC1**ALW OR SM-MAC-G2

TCS\_153 All EF structures shall be transparent.

TCS\_154 The driver card application generation 2 shall have the following data structure:

<b>File / Data element</b>	<i>No of Records</i>	<i>Size (bytes)</i>		<i>Default Values</i>
		<b>M in</b>	<b>M ax</b>	
D Tachograph_G2	19	39		
E Application_Identification	15	15		
DriverCardApplicationIdentificat	15	15		
typeOfTachographCardId	1	1	{00}	
cardStructureVersion	2	2	{00}	
noOfEventsPerType	1	1	{00}	
noOfFaultsPerType	1	1	{00}	
activityStructureLength	2	2	{00}	
noOfCardVehicleRecords	2	2	{00}	
noOfCardPlaceRecords	2	2	{00}	
noOfGNSSCDRecords	2	2	{00}	

File / Data element	No of Re cor ds	Size (bytes)		Default Values	
		M in	M ax		
noOfSpecificConditionRecord	2	2		{00}	
E CardMA_Certificate	20	34			
CardMACertificate	20	34		{00..	
E CardSignCertificate	20	34			
CardSignCertificate	20	34		{00..	
E CA_Certificate	20	34			
MemberStateCertificate	20	34		{00..	
E Link_Certificate	20	34			
LinkCertificate	20	34		{00..	
E Identification			143	143	
CardIdentification			65	65	
cardIssuingMemberState			1	1	{00}
cardNumber			16	16	{20..2
cardIssuingAuthorityName			36	36	{20..2
cardIssueDate			4	4	{00..0
cardValidityBegin			4	4	{00..0
cardExpiryDate			4	4	{00..0
DriverCardHolderIdentification			78	78	
cardHolderName			72	72	
holderSurname			36	36	{00,
holderFirstNames			36	36	{00,
cardHolderBirthDate			4	4	{00..0
cardHolderPreferredLanguage			2	2	{20
EF Card_Download			4	4	
LastCardDownload			4	4	
EF Driving_Licence_Info			53	53	
CardDrivingLicenceInformation			53	53	
drivingLicenceIssuingAuthority			36	36	{00,
drivingLicenceIssuingNation			1	1	{00}
drivingLicenceNumber			16	16	{20..2
EF Events_Data			158	316	
CardEventData			158	316	
cardEventRecords		11	144	288	
CardEventRecord		n <sub>1</sub>	24	24	
eventType			1	1	{00}
eventBeginTime			4	4	{00..0
eventEndTime			4	4	{00..0
eventVehicleRegistration					
vehicleRegistrationNation			1	1	{00}
vehicleRegistrationNumber			14	14	{00,
EF Faults_Data			576	115	
CardFaultData			576	115	
cardFaultRecords		2	288	576	
CardFaultRecord		n <sub>2</sub>	24	24	

File / Data element	No of Re cor ds	Size (bytes)		Default Values
		M in	M ax	
faultType		1	1	{00}
faultBeginTime		4	4	{00..0}
faultEndTime		4	4	{00..0}
faultVehicleRegistration				
vehicleRegistrationNation		1	1	{00}
vehicleRegistrationNumber		14	14	{00, 20..20 }
EF Driver_Activity_Data		554	137	
CardDriverActivity		554	137	
activityPointerOldestDayRecord		2	2	{00}
activityPointerNewestRecord		2	2	{00}
activityDailyRecords	n <sub>6</sub>	554	137	{00..0}
EF Vehicles_Used		403	960	
CardVehiclesUsed		403	960	
vehiclePointerNewestRecord		2	2	{00}
cardVehicleRecords		403	960	
CardVehicleRecord	n <sub>3</sub>	48	48	
vehicleOdometerBegin		3	3	{00..0}
vehicleOdometerEnd		3	3	{00..0}
vehicleFirstUse		4	4	{00..0}
vehicleLastUse		4	4	{00..0}
vehicleRegistration				
vehicleRegistrationNation		1	1	{00}
vehicleRegistrationNumber		14	14	{00, 20..20 }
vuDataBlockCounter		2	2	{00}
vehicleIdentificationNumber		17	17	{20..2}
E Places		176	235	
CardPlaceDailyWorkPeriod		176	235	
placePointerNewestRecord		6	4	
placeRecords		176	235	
PlaceRecord	n <sub>4</sub>	21	21	
entryTime		4	4	{00..0}
entryTypeDailyWorkPeriod		1	1	{00}
dailyWorkPeriodCountry		1	1	{00}
dailyWorkPeriodRegion		1	1	{00}
vehicleOdometerValue		3	3	{00..0}
entryGNSSPlaceRecord		11	11	
timeStamp		4	4	{00..0}
gnssAccuracy		1	1	{00}
geoCoordinates		6	6	{00..0}
E Current_Usage		19	19	
CardCurrentUse		19	19	



File / Data element	No of Re cor ds	Size (bytes)		Default Values
		M in	M ax	
sessionOpenVehicle				n1
vehicleRegistrationNation		1	1	{00}
vehicleRegistrationNumber		14	14	{00,
E Control_Activity_Data		46	46	
CardControlActivityDataRecord		46	46	
controlType		1	1	{00}
controlTime		4	4	{00..0
controlCardNumber				
cardType		1	1	{00}
cardIssuingMemberState		1	1	{00}
cardNumber		16	16	{20..2
controlVehicleRegistration				
vehicleRegistrationNation		1	1	{00}
vehicleRegistrationNumber		14	14	{00,
controlDownloadPeriodBegin		4	4	{00..0
controlDownloadPeriodEnd		4	4	{00..0
E Specific_Conditions		282	562	
SpecificConditions		282	562	
conditionPointerNewestRecord		2	2	{00
specificConditionRecords		280	560	
SpecificConditionRecord	n9	5	5	
entryTime		4	4	{00..0
specificConditionType		1	1	{00}
E VehicleUnits_Used		842	200	
CardVehicleUnitsUsed		842	200	
vehicleUnitPointerNewestRecord		2	2	{00
cardVehicleUnitRecords		840	200	
CardVehicleUnitRecord	n7	10	10	
timeStamp		4	4	{00..0
manufacturerCode		1	1	{00}
deviceID		1	1	{00}
vuSoftwareVersion		4	4	{00..0
E GNSS_Places		378	504	
GNSSContinuousDriving		378	504	
gnssCDPointerNewestRecord		2	2	{00
gnssContinuousDrivingRecords		378	504	{00}
GNSSContinuousDrivingRecord	n8	15	15	
timeStamp		4	4	{00..0
gnssPlaceRecord		11	11	
timeStamp		4	4	{00..0
gnssAccuracy		1	1	{00}
geoCoordinates		6	6	{00..0

TCS\_155 The following values, used to provide sizes in the table above, are the minimum and maximum record number values the driver card data structure must use for a generation 2 application:

	<i>Min</i>	<i>Max</i>
n NoOfEventsPerType 1	6	12
n NoOfFaultsPerType 2	12	24
n NoOfCardVehicleRecords 3	84	200
n NoOfCardPlaceRecords 4	84	112
n CardActivityLengthRange 6	5544 bytes (28 days * 93 activity changes)	13776 Bytes (28 days * 240 activity changes)
n NoOfCardVehicleUnitRecords 7	84	200
n NoOfGNSSCDRecords 8	252	336
n NoOfSpecificConditionRecords 9	56	112

4.3. Workshop card applications

4.3.1 Workshop card application generation 1

TCS\_156 After its personalisation, the workshop card application generation 1 shall have the following permanent file structure and file access rules:

File	<i>Access rules</i>			
	File ID	Read	Select	Update
DF Tachograph	'050		SC1	
EF Application_Identification	'050	SC2	SC1	NEV
EF Card_Certificate	'C1	SC2	SC1	NEV
EF CA_Certificate	'C1	SC2	SC1	NEV
EF Identification	'052	SC2	SC1	NEV
EF Card_Download	'050	SC2	SC1	<b>SC1</b>
EF Calibration	'050	SC2	SC1	SC3
EF Sensor_Installation_Data	'050	<b>SC4</b>	SC1	NEV
EF Events_Data	'050	SC2	SC1	SC3
EF Faults_Data	'050	SC2	SC1	SC3
EF Driver_Activity_Data	'050	SC2	SC1	SC3
EF Vehicle_Information	'050	SC2	SC1	SC3

					<i>Access rules</i>
	5h'				
EF Places	'050	SC2	SC1	SC3	
EF Current_Usage	'050	SC2	SC1	SC3	
EF Control_Activity_Data	'050	SC2	SC1	SC3	
EF Specific_Conditions	'052	SC2	SC1	SC3	

The following abbreviations for the security conditions are used in this table:

**SC1** ALW OR SM-MAC-G2

**SC2** ALW OR SM-MAC-G1 OR SM-MAC-G2

**SC3** SM-MAC-G1 OR SM-MAC-G2

- SC4** For the READ BINARY command with even INS byte:  
(PLAIN-C AND SM-R-ENC-G1) OR (SM-C-MAC-G1 AND SM-R-ENC-MAC-G1) OR  
(SM-C-MAC-G2 AND SM-R-ENC-MAC-G2)  
For the READ BINARY command with odd INS byte (if supported): NEV

TCS\_157 All EF structures shall be transparent.

TCS\_158 The workshop card application generation 1 shall have the following data structure:

<b>File / Data element</b>	<i>No of Records</i>	<i>Size (Bytes)</i>		<i>Default Values</i>
		<b>Min</b>	<b>Max</b>	
DF Tachograph		11055	29028	
EF Application_Identification		11	11	
WorkshopCardApplicationIdentification		11	11	
typeOfTachographCardId		1	1	{00}
cardStructureVersion		2	2	{00 00}
noOfEventsPerType		1	1	{00}
noOfFaultsPerType		1	1	{00}
activityStructureLength		2	2	{00 00}
noOfCardVehicleRecords		2	2	{00 00}
noOfCardPlaceRecords		1	1	{00}
noOfCalibrationRecords		1	1	{00}
EF Card_Certificate		194	194	
CardCertificate		194	194	{00..00}
EF CA_Certificate		194	194	
MemberStateCertificate		194	194	{00..00}
EF Identification		211	211	
CardIdentification		65	65	
cardIssuingMemberState		1	1	{00}
cardNumber		16	16	{20..20}
cardIssuingAuthorityName		36	36	{00,
cardIssueDate		4	4	{00..00}
cardValidityBegin		4	4	{00..00}
cardExpiryDate		4	4	{00..00}

File / Data element	No of Records	Size (Bytes)		Default Values
		Min	Max	
workshopName		36	36	{00,
workshopAddress		36	36	{00,
cardHolderName				
holderSurname		36	36	{00,
holderFirstNames		36	36	{00,
cardHolderPreferredLanguage		2	2	{20 20}
EF Card_Download		2	2	
NoOfCalibrationsSinceDownload		2	2	{00 00}
EF Calibration		9243	26778	
WorkshopCardCalibrationData		9243	26778	
calibrationTotalNumber		2	2	{00 00}
calibrationPointerNewestRecord		1	1	{00}
calibrationRecords		9240	26775	
WorkshopCardCalibrationRecord	n <sub>5</sub>	105	105	
calibrationPurpose		1	1	{00}
vehicleIdentificationNumber		17	17	{20..20}
vehicleRegistration				
vehicleRegistrationNation		1	1	{00}
vehicleRegistrationNumber		14	14	{00,
wVehicleCharacteristicConstant		2	2	{00 00}
kConstantOfRecordingEquipment		2	2	{00 00}
lTyreCircumference		2	2	{00 00}
tyreSize		15	15	{20..20}
authorisedSpeed		1	1	{00}
oldOdometerValue		3	3	{00..00}
newOdometerValue		3	3	{00..00}
oldTimeValue		4	4	{00..00}
newTimeValue		4	4	{00..00}
nextCalibrationDate		4	4	{00..00}
vuPartNumber		16	16	{20..20}
vuSerialNumber		8	8	{00..00}
sensorSerialNumber		8	8	{00..00}
EF Sensor_Installation_Data		16	16	
SensorInstallationSecData		16	16	{00..00}
EF Events_Data		432	432	
CardEventData		432	432	
cardEventRecords	6	72	72	
CardEventRecord	n <sub>1</sub>	24	24	
eventType		1	1	{00}
eventBeginTime		4	4	{00..00}
eventEndTime		4	4	{00..00}
eventVehicleRegistration				
vehicleRegistrationNation		1	1	{00}
vehicleRegistrationNumber		14	14	{00,

File / Data element	No of Records	Size (Bytes)		Default Values
		Min	Max	
EF Faults_Data		288	288	
CardFaultData		288	288	
cardFaultRecords	2	144	144	
CardFaultRecord	n <sub>2</sub>	24	24	
faultType		1	1	{00}
faultBeginTime		4	4	{00..00}
faultEndTime		4	4	{00..00}
faultVehicleRegistration				
vehicleRegistrationNation		1	1	{00}
vehicleRegistrationNumber		14	14	{00,
EF Driver_Activity_Data		202	496	
CardDriverActivity		202	496	
activityPointerOldestDayRecord		2	2	{00 00}
activityPointerNewestRecord		2	2	{00 00}
activityDailyRecords	n <sub>6</sub>	198	492	{00..00}
EF Vehicles_Used		126	250	
CardVehiclesUsed		126	250	
vehiclePointerNewestRecord		2	2	{00 00}
cardVehicleRecords		124	248	
CardVehicleRecord	n <sub>3</sub>	31	31	
vehicleOdometerBegin		3	3	{00..00}
vehicleOdometerEnd		3	3	{00..00}
vehicleFirstUse		4	4	{00..00}
vehicleLastUse		4	4	{00..00}
vehicleRegistration				
vehicleRegistrationNation		1	1	{00}
vehicleRegistrationNumber		14	14	{00,
vuDataBlockCounter		2	2	{00 00}
EF Places		61	81	
CardPlaceDailyWorkPeriod		61	81	
placePointerNewestRecord		1	1	{00}
placeRecords		60	80	
PlaceRecord	n <sub>4</sub>	10	10	
entryTime		4	4	{00..00}
entryTypeDailyWorkPeriod		1	1	{00}
dailyWorkPeriodCountry		1	1	{00}
dailyWorkPeriodRegion		1	1	{00}
vehicleOdometerValue		3	3	{00..00}
EF Current_Usage		19	19	
CardCurrentUse		19	19	
sessionOpenTime		4	4	{00..00}
sessionOpenVehicle				
vehicleRegistrationNation		1	1	{00}
vehicleRegistrationNumber		14	14	{00,
EF Control_Activity_Data		46	46	

File / Data element	No of Records	Size (Bytes)		Default Values
		Min	Max	
controlType		1	1	{00}
controlTime		4	4	{00..00}
controlCardNumber				
cardType		1	1	{00}
cardIssuingMemberState		1	1	{00}
cardNumber		16	16	{20..20}
controlVehicleRegistration				
vehicleRegistrationNation		1	1	{00}
vehicleRegistrationNumber		14	14	{00,
controlDownloadPeriodBegin		4	4	{00..00}
controlDownloadPeriodEnd		4	4	{00..00}
EF Specific_Conditions		10	10	
SpecificConditionRecord	2	5	5	
entryTime		4	4	{00..00}
SpecificConditionType		1	1	{00}

TCS\_159 The following values, used to provide sizes in the table above, are the minimum and maximum record number values the workshop card data structure must use for a generation 1 application:

	<i>Min</i>	<i>Max</i>
n NoOfEventsPerType 1	3	3
n NoOfFaultsPerType 2	6	6
n NoOfCardVehicleRecords 3	4	8
n NoOfCardPlaceRecords 4	6	8
n NoOfCalibrationRecords 5	88	255
n CardActivityLengthRange 6	198 bytes (1 day *)	492 bytes (1 day *)
	93 activity changes)	240 activity changes)

#### 4.3.2 Workshop card application generation 2

TCS\_160 After its personalisation, the workshop card application generation 2 shall have the following permanent file structure and file access rules.

Note: .The short EF identifier SFID is given as decimal number, e.g. the value 30 corresponds to 11110 in binary.

File	Access rules				
	File ID	SF ID	Read	Select	Update
DF Tachograph_G2			SC	SC1	
EF Application_Identification	'0501	1	SC	SC1	NEV
EF CardMA_Certificate	'C10	2	SC	SC1	NEV
EF CardSignCertificate	'C10	3	SC	SC1	NEV
EF CA_Certificate	'C10	4	SC	SC1	NEV
EF Link_Certificate	'C10	5	SC	SC1	NEV
EF Identification	'0520	6	SC	SC1	NEV
EF Card_Download	'0509	7	SC	SC1	SC1
EF Calibration	'050	10	SC	SC1	SM-
EF Sensor_Installation_Data	'050	11	<b>SC</b>	SM-	NEV
EF Events_Data	'0502	12	SC	SC1	SM-
EF Faults_Data	'0503	13	SC	SC1	SM-
EF Driver_Activity_Data	'0504	14	SC	SC1	SM-
EF Vehicles_Used	'0505	15	SC	SC1	SM-
EF Places	'0506	16	SC	SC1	SM-
EF Current_Usage	'0507	17	SC	SC1	SM-
EF Control_Activity_Data	'0508	18	SC	SC1	SM-
EF Specific_Conditions	'0522	19	SC	SC1	SM-
EF VehicleUnits_Used	'0523	20	SC	SC1	SM-
EF GNSS_Places	'0524	21	SC	SC1	SM-

The following abbreviations for the security conditions are used in this table:

**SC1** ALW OR SM-MAC-G2

**SC5** For the Read Binary command with even INS byte: SM-C-MAC-G2 AND SM-R-ENC-MAC-G2

For the Read Binary command with odd INS byte (if supported): NEV

TCS\_161 All EFs structures shall be transparent.

TCS\_162 The workshop card application generation 2 shall have the following data structure:

File / Data element	No of Reco rds	Size (Bytes)		Default Values
		Mi n	Ma x	
DF Tachograph_G2		17	471	
EF Application_Identification		17	17	
WorkshopCardApplicationIdentification		17	17	
typeOfTachographCardId		1	1	{00}
cardStructureVersion		2	2	{00}
noOfEventsPerType		1	1	{00}
noOfFaultsPerType		1	1	{00}
activityStructureLength		2	2	{00}
noOfCardVehicleRecords		2	2	{00}
noOfCardPlaceRecords		2	2	{00}
noOfCalibrationRecords		2	2	{00}
noOfGNSSCDRecords		2	2	{00..00}
noOfSpecificConditionRecords		2	2	{00..00}
EF CardMA_Certificate		20	341	
CardMACertificate		20	341	{00..00}
EF CardSignCertificate		20	341	
CardSignCertificate		20	341	{00..00}
EF CA_Certificate		20	341	
MemberStateCertificate		20	341	{00..00}
EF Link_Certificate		20	341	
LinkCertificate		20	341	{00..00}
EF Identification		21	211	
CardIdentification		65	65	
cardIssuingMemberState		1	1	{00}
cardNumber		16	16	{20..20}
cardIssuingAuthorityName		36	36	{00,
cardIssueDate		4	4	{00..00}
cardValidityBegin		4	4	{00..00}
cardExpiryDate		4	4	{00..00}
WorkshopCardHolderIdentification		14	146	
workshopName		36	36	{00,
workshopAddress		36	36	{00,
cardHolderName				
holderSurname		36	36	{00,
holderFirstNames		36	36	{00,
cardHolderPreferredLanguage		2	2	{20}
EF Card_Download		2	2	
NoOfCalibrationsSinceDownload		2	2	{00}
EF Calibration		14	428	
WorkshopCardCalibrationData		14	428	
calibrationTotalNumber		2	2	{00}
calibrationPointerNewestRecord		2	2	{00}
calibrationRecords		14	428	
WorkshopCardCalibrationRecord	n5	16	168	
calibrationPurpose		1	1	{00}



File / Data element	No of Records	Size (Bytes)		Default Values
		Min	Max	
vehicleIdentificationNumber		17	17	{20..20}
vehicleRegistration				
vehicleRegistrationNation		1	1	{00}
vehicleRegistrationNumber		14	14	{00,
wVehicleCharacteristicConstant		2	2	{00
kConstantOfRecordingEquipmen		2	2	{00
lTyreCircumference		2	2	{00
tyreSize		15	15	{20..20
authorisedSpeed		1	1	{00}
oldOdometerValue		3	3	{00..00
newOdometerValue		3	3	{00..00
oldTimeValue		4	4	{00..00
newTimeValue		4	4	{00..00
nextCalibrationDate		4	4	{00..00
vuPartNumber		16	16	{20..20
vuSerialNumber		8	8	{00..00
sensorSerialNumber		8	8	{00..00
sensorGNSSSerialNumber		8	8	{00..00
rcmSerialNumber		8	8	{00..00
vuAbility		1	1	{00}
sealDataCard		46	46	
noOfSealRecords		1	1	{00}
SealRecords		45	45	
└ SealRecord	5	9	9	
└ equipmentType		1	1	{00}
└ extendedSealIdentifier		8	8	{00..00 }
EF Sensor_Installation_Data		18	102	
SensorInstallationSecData		18	102	{00..00
EF Events_Data		79	792	
CardEventData		79	792	
cardEventRecords	11	72	72	
CardEventRecord	n <sub>1</sub>	24	24	
eventType		1	1	{00}
eventBeginTime		4	4	{00..00
eventEndTime		4	4	{00..00
eventVehicleRegistration				
vehicleRegistrationNation		1	1	{00}
vehicleRegistrationNumber		14	14	{00,
EF Faults_Data		28	288	
CardFaultData		28	288	
cardFaultRecords	2	14	144	
CardFaultRecord	n <sub>2</sub>	24	24	

File / Data element	No of Reco rds	Size (Bytes)		Default Values
		Mi n	Ma x	
faultBeginTime		4	4	{00..00
faultEndTime		4	4	{00..00
faultVehicleRegistration				
vehicleRegistrationNation		1	1	{00}
vehicleRegistrationNumber		14	14	{00,
EF Driver_Activity_Data		20	496	
CardDriverActivity		20	496	
activityPointerOldestDayRecord		2	2	{00
activityPointerNewestRecord		2	2	{00
activityDailyRecords	n <sub>6</sub>	19	492	{00..00
EF Vehicles_Used		19	386	
CardVehiclesUsed		19	386	
vehiclePointerNewestRecord		2	2	{00
cardVehicleRecords		19	384	
CardVehicleRecord	n <sub>3</sub>	48	48	
vehicleOdometerBegin		3	3	{00..00
vehicleOdometerEnd		3	3	{00..00
vehicleFirstUse		4	4	{00..00
vehicleLastUse		4	4	{00..00
vehicleRegistration				
vehicleRegistrationNation		1	1	{00}
vehicleRegistrationNumber		14	14	{00,
vuDataBlockCounter		2	2	{00
vehicleIdentificationNumber		17	17	{20..20
EF Places		12	170	
CardPlaceDailyWorkPeriod		12	170	
placePointerNewestRecord		2	2	{00
placeRecords		12	168	
PlaceRecord	n <sub>4</sub>	21	21	
entryTime		4	4	{00..00
entryTypeDailyWorkPeriod		1	1	{00}
dailyWorkPeriodCountry		1	1	{00}
dailyWorkPeriodRegion		1	1	{00}
vehicleOdometerValue		3	3	{00..00
entryGNSSPlaceRecord		11	11	{00..00
timeStamp		4	4	{00..00
gnssAccuracy		1	1	{00}
geoCoordinates		6	6	{00..00
EF Current_Usage		19	19	
CardCurrentUse		19	19	
sessionOpenTime		4	4	{00..00
sessionOpenVehicle				
vehicleRegistrationNation		1	1	{00}
vehicleRegistrationNumber		14	14	{00,

File / Data element	No of Reco rds	Size (Bytes)		Default Values
		Mi n	Ma x	
EF Control_Activity_Data		46	46	
CardControlActivityDataRecord		46	46	
controlType		1	1	{00}
controlTime		4	4	{00..00}
controlCardNumber				
cardType		1	1	{00}
cardIssuingMemberState		1	1	{00}
cardNumber		16	16	{20..20}
controlVehicleRegistration				
vehicleRegistrationNation		1	1	{00}
vehicleRegistrationNumber		14	14	{00,
controlDownloadPeriodBegin		4	4	{00..00}
controlDownloadPeriodEnd		4	4	{00..00}
EF VehicleUnits_Used		42	42	
CardVehicleUnitsUsed		42	82	
vehicleUnitPointerNewestRecord		2	2	{00}
cardVehicleUnitRecords		40	80	
CardVehicleUnitRecord	n <sub>7</sub>	10	10	
timeStamp		4	4	{00..00}
manufacturerCode		1	1	{00..00}
deviceID		1	1	{00..00}
vuSoftwareVersion		4	4	{00..00}
EF GNSS_Places		26	362	
GNSSContinuousDriving		26	362	
gnssCDPointerNewestRecord		2	2	{00}
gnssContinuousDrivingRecords		26	360	
GNSSContinuousDrivingRecord	n <sub>8</sub>	15	15	
timeStamp		4	4	{00..00}
gnssPlaceRecord		11	11	
timeStamp		4	4	{00..00}
gnssAccuracy		1	1	{00}
geoCoordinates		6	6	{00..00}
EF Specific_Conditions		12	22	
SpecificConditions		12	22	
conditionPointerNewestRecord		2	2	{00}
specificConditionRecords		10	20	
SpecificConditionRecord	n <sub>9</sub>	5	5	
entryTime		4	4	{00..00}
specificConditionType		1	1	{00}

TCS\_163 The following values, used to provide sizes in the table above, are the minimum and maximum record number values the workshop card data structure must use for a generation 2 application:

	<i>Min</i>	<i>Max</i>
n NoOfEventsPerType 1	3	3
n NoOfFaultsPerType 2	6	6
n NoOfCardVehicleRecords 3	4	8
n NoOfCardPlaceRecords 4	6	8
n NoOfCalibrationRecords 5	88	255
n CardActivityLengthRange 6	198 bytes (1 day * 93 activity changes)	492 bytes (1 day * 240 activity changes)
n NoOfCardVehicleUnitRecords 7	4	8
n NoOfGNSSCDRecords 8	18	24
n NoOfSpecificConditionRecords 9	2	4

#### 4.4. Control card applications

##### 4.4.1 Control Card application generation 1

TCS\_164 After its personalisation, the control card application generation 1 shall have the following permanent file structure and file access rules:

<b>File</b>	<i>Access rules</i>			
	<b>File ID</b>	<b>Read</b>	<b>Select</b>	<b>Update</b>
DF Tachograph	'050			
EF Application_Identification	'050	SC2	SC1	NEV
EF Card_Certificate	'C1	SC2	SC1	NEV
EF CA_Certificate	'C1	SC2	SC1	NEV
EF Identification	'052	<b>SC6</b>	SC1	NEV
EF Controller_Activity_Data	'050	SC2	SC1	SC3

The following abbreviations for the security conditions are used in this table:

**SC1**ALW OR SM-MAC-G2

SC2ALW OR SM-MAC-G1 OR SM-MAC-G2

SC3SM-MAC-G1 OR SM-MAC-G2

SC6EXT-AUT-G1 OR SM-MAC-G1 OR SM-MAC-G2

TCS\_165 All EF structures shall be transparent.

TCS\_166 The control card application generation 1 shall have the following data structure:

File / Data element	No of Reco rds	Size (Bytes)		
		Min	Max	
DF Tachograph		111	245	
EF Application_Identification		5	5	
ControlCardApplicationIdentification		5	5	
typeOfTachographCardId		1	1	{00}
cardStructureVersion		2	2	{00}
noOfControlActivityRecords		2	2	{00}
EF Card_Certificate		194	194	
CardCertificate		194	194	{00..00}
EF CA_Certificate		194	194	
MemberStateCertificate		194	194	{00..00}
EF Identification		211	211	
CardIdentification		65	65	
cardIssuingMemberState		1	1	{00}
cardNumber		16	16	{20..20}
cardIssuingAuthorityName		36	36	{00,
cardIssueDate		4	4	{00..00}
cardValidityBegin		4	4	{00..00}
cardExpiryDate		4	4	{00..00}
ControlCardHolderIdentification		146	146	
controlBodyName		36	36	{00,
controlBodyAddress		36	36	{00,
cardHolderName				
holderSurname		36	36	{00,
holderFirstNames		36	36	{00,
cardHolderPreferredLanguage		2	2	{20}
EF Controller_Activity_Data		105	239	
ControlCardControlActivityData		105	239	
controlPointerNewestRecord		2	2	{00}
controlActivityRecords		105	239	
controlActivityRecord	n7	46	46	
controlType		1	1	{00}
controlTime		4	4	{00..00}
controlledCardNumber				
cardType		1	1	{00}
cardIssuingMemberState		1	1	{00}
cardNumber		16	16	{20..20}

File / Data element	No of Records	Size (Bytes)	
		Min	Max
vehicleRegistrationNation	1	1	{00}
vehicleRegistrationNumber	14	14	{00,
controlDownloadPeriodBegin	4	4	{00..00
controlDownloadPeriodEnd	4	4	{00..00

TCS\_167 The following values, used to provide sizes in the table above, are the minimum and maximum record number values the control card data structure must use for a generation 1 application:

	Min	Max
n NoOfControlActivityRecords	230	520

4.4.2 Control card application generation 2

TCS\_168 After its personalisation, the control card application generation 2 shall have the following permanent file structure and file access rules.

Note: .The short EF identifier SFID is given as decimal number, e.g. the value 30 corresponds to 11110 in binary.

File	File ID	SF ID	Access rules	
			Read / Select	Update
DF Tachograph_G2			SC1	
EF Application_Identification	'0501	1	SC1	NEV
EF CardMA_Certificate	'C10	2	SC1	NEV
EF CA_Certificate	'C10	4	SC1	NEV
EF Link_Certificate	'C10	5	SC1	NEV
EF Identification	'0520	6	SC1	NEV
EF Controller_Activity_Data	'050	14	SC1	SM-MAC-

The following abbreviation for the security condition is used in this table:

SC1ALW OR SM-MAC-G2

TCS\_169 All EF structures shall be transparent.

TCS\_170 The control card application generation2 shall have the following data structure:

File / Data element	No of Reco rds	Size (Bytes)		
		Min	Max	
DF Tachograph_G2		114	251	
EF Application_Identification		5	5	
ControlCardApplicationIdentification		5	5	
typeOfTachographCardId		1	1	{00}
cardStructureVersion		2	2	{00}
noOfControlActivityRecords		2	2	{00}
EF CardMA_Certificate		204	341	
CardMACertificate		204	341	{00..0}
EF CA_Certificate		204	341	
MemberStateCertificate		204	341	{00..0}
EF Link_Certificate		204	341	
LinkCertificate		204	341	{00..0}
EF Identification		211	211	
CardIdentification		65	65	
cardIssuingMemberState		1	1	{00}
cardNumber		16	16	{20..2}
cardIssuingAuthorityName		36	36	{00,
cardIssueDate		4	4	{00..0}
cardValidityBegin		4	4	{00..0}
cardExpiryDate		4	4	{00..0}
ControlCardHolderIdentification		146	146	
controlBodyName		36	36	{00,
controlBodyAddress		36	36	{00,
cardHolderName				
holderSurname		36	36	{00,
holderFirstNames		36	36	{00,
cardHolderPreferredLanguage		2	2	{20}
EF Controller_Activity_Data		105	239	
ControlCardControlActivityData		105	239	
controlPointerNewestRecord		2	2	{00}
controlActivityRecords		105	239	
controlActivityRecord	n7	46	46	
controlType		1	1	{00}
controlTime		4	4	{00..0}
controlledCardNumber				
cardType		1	1	{00}
cardIssuingMemberState		1	1	{00}
cardNumber		16	16	{20..2}
controlledVehicleRegistration				
vehicleRegistrationNation		1	1	{00}
vehicleRegistrationNumber		14	14	{00,
controlDownloadPeriodBegin		4	4	{00..0}
controlDownloadPeriodEnd		4	4	{00..0}

TCS\_171 The following values, used to provide sizes in the table above, are the minimum and maximum record number values the control card data structure must use for a generation 2 application:

	<i>Min</i>	<i>Max</i>
n NoOfControlActivityRecords 7	230	520

4.5. Company card applications

4.5.1 Company card application generation 1

TCS\_172 After its personalisation, the company card application generation 1 shall have the following permanent file structure and file access rules:

File	<i>Access rules</i>			
	File ID	Read	Select	Update
DF Tachograph	'050		SC1	
EF Application_Identification	'050	SC2	SC1	NEV
EF Card_Certificate	'C1	SC2	SC1	NEV
EF CA_Certificate	'C1	SC2	SC1	NEV
EF Identification	'052	<b>SC6</b>	SC1	NEV
EF Company_Activity_Data	'050	SC2	SC1	SC3

The following abbreviations for the security conditions are used in this table:

**SC1**ALW OR SM-MAC-G2

**SC2**ALW OR SM-MAC-G1 OR SM-MAC-G2

**SC3**SM-MAC-G1 OR SM-MAC-G2

**SC6**EXT-AUT-G1 OR SM-MAC-G1 OR SM-MAC-G2

TCS\_173 All EF structures shall be transparent.

TCS\_174 The company card application generation 1 shall have the following data structure:

File / Data element	No of Records	Size (bytes)		Default Values
		Min	Max	
DF Tachograph		111	244	
EF Application_Identification		5	5	
CompanyCardApplicationIdentification		5	5	
typeOfTachographCardId		1	1	{00}
cardStructureVersion		2	2	{00}
noOfCompanyActivityRecords		2	2	{00}
EF Card_Certificate		194	194	
CardCertificate		194	194	{00..00}



File / Data element	No of Reco rds	Size (bytes)		Default Values
		Min	Ma x	
EF CA_Certificate		194	194	
MemberStateCertificate		194	194	{00..00}
EF Identification		139	139	
CardIdentification		65	65	
cardIssuingMemberState		1	1	{00}
cardNumber		16	16	{20..20}
cardIssuingAuthorityName		36	36	{00,
cardIssueDate		4	4	{00..00}
cardValidityBegin		4	4	{00..00}
cardExpiryDate		4	4	{00..00}
CompanyCardHolderIdentification		74	74	
companyName		36	36	{00,
companyAddress		36	36	{00,
cardHolderPreferredLanguage		2	2	{20}
EF Company_Activity_Data		105	239	
CompanyActivityData		105	239	
companyPointerNewestRecord		2	2	{00}
companyActivityRecords		105	239	
companyActivityRecord	ns	46	46	
companyActivityType		1	1	{00}
companyActivityTime		4	4	{00..00}
cardNumberInformation				
cardType		1	1	{00}
cardIssuingMemberState		1	1	{00}
cardNumber		16	16	{20..20}
vehicleRegistrationInformation				
vehicleRegistrationNation		1	1	{00}
vehicleRegistrationNumber		14	14	{00,
downloadPeriodBegin		4	4	{00..00}
downloadPeriodEnd		4	4	{00..00}

TCS\_175 The following values, used to provide sizes in the table above, are the minimum and maximum record number values the company card data structure must use for a generation 1 application:

	Min	Max
n NoOfCompanyActivityRecords	230	520
8		

#### 4.5.2 Company card application generation 2

TCS\_176 After its personalisation, the company card application generation 2 shall have the following permanent file structure and file access rules.

Note: .The short EF identifier SFID is given as decimal number, e.g. the value 30 corresponds to 11110 in binary.

<i>Access rules</i>				
<b>File</b>	<b>File ID</b>	<b>SF ID</b>	<b>Read / Select</b>	<b>Update</b>
DF Tachograph_G2			SC1	
EF Application_Identification	'0501	1	SC1	NEV
EF CardMA_Certificate	'C10	2	SC1	NEV
EF CA_Certificate	'C10	4	SC1	NEV
EF Link_Certificate	'C10	5	SC1	NEV
EF Identification	'0520	6	SC1	NEV
EF Company_Activity_Data	'050	14	SC1	SM-MAC-

The following abbreviation for the security condition is used in this table:

**SC1ALW OR SM-MAC-G2**

TCS\_177 All EF structures shall be transparent.

TCS\_178 The company card application generation 2 shall have the following data structure:

<i>File / Data element</i>	<i>No of Reco rds</i>	<i>Size (bytes)</i>		<i>Default Values</i>
		<b>Min</b>	<b>Max</b>	
DF Tachograph_G2		113	250	
EF Application_Identification		5	5	
CompanyCardApplicationIdentification		5	5	
typeOfTachographCardId		1	1	{00}
cardStructureVersion		2	2	{00}
noOfCompanyActivityRecords		2	2	{00}
EF CardMA_Certificate		204	341	
CardMACertificate		204	341	{00..00}
EF CA_Certificate		204	341	
MemberStateCertificate		204	341	{00..00}
EF Link_Certificate		204	341	
LinkCertificate		204	341	{00..00}
EF Identification		139	139	
CardIdentification		65	65	
cardIssuingMemberState		1	1	{00}
cardNumber		16	16	{20..20}
cardIssuingAuthorityName		36	36	{00,
cardIssueDate		4	4	{00..00}
cardValidityBegin		4	4	{00..00}
cardExpiryDate		4	4	{00..00}
CompanyCardHolderIdentification		74	74	
companyName		36	36	{00,
companyAddress		36	36	{00,

File / Data element	No of Reco rds	Size (bytes)		Default Values
		Min	Max	
cardHolderPreferredLanguage	2	2		{20 20}
EF Company_Activity_Data		105	239	
CompanyActivityData		105	239	
companyPointerNewestRecord	2	2		{00}
companyActivityRecords		105	239	
companyActivityRecord	ns	46	46	
companyActivityType	1	1		{00}
companyActivityTime	4	4		{00..00}
cardNumberInformation				
cardType	1	1		{00}
cardIssuingMemberState	1	1		{00}
cardNumber	16	16		{20..20}
vehicleRegistrationInformation				
vehicleRegistrationNation	1	1		{00}
vehicleRegistrationNumber	14	14		{00,
downloadPeriodBegin	4	4		{00..00}
downloadPeriodEnd	4	4		{00..00}

























TCS\_179 The following values, used to provide sizes in the table above, are the minimum and maximum record number values the company card data structure must use for a generation 2 application:

	Min	Max
n NoOfCompanyActivityRecords	230	520
8		

## Appendix 3. Pictograms

PIC\_001 The tachograph may optionally use the following pictograms and pictogram combinations (or pictograms and combination similar enough to be unambiguously identifiable with these):

### 5. Basic pictograms

	<b><u>People</u></b>	<b><u>Actions</u></b>	<b><u>Modes of operation</u></b>
	Company		Company mode
	Controller	Control	Control mode
	Driver	Driving	Operational mode
	Workshop/test station	Inspection/calibration	Calibration mode
	Manufacturer		
	<b><u>Activities</u></b>	<b><u>Duration</u></b>	
	Available	Current availability period	
	Driving	Continuous driving time	
	Rest	Current rest period	
	Other Work	Current work period	
	Break	Cumulative break time	
	Unknown		
	<b><u>Equipment</u></b>	<b><u>Functions</u></b>	
	Driver slot		
	Co-driver slot		
	Card		
	Clock		
	Display	Displaying	
	External storage	Downloading	
	Power supply		
	Printer/printout	Printing	
	Sensor		
	Tyre size		
	Vehicle/vehicle unit		
	GNSS facility		
	Remote Detection facility		

☞ ITS interface

### Specific conditions

OUT Out of scope

⚓ Ferry/train crossing

### Miscellaneous

! Events

✕ Faults

▶ Start of daily work period

◀ End of daily work period

• Location

M Manual entry of driver activities

🔒 Security

➤ Speed

🕒 Time

Σ Total/summary

### Qualifiers

24h Daily

| Weekly

|| Two weeks

➔ From or to

## 6. Pictogram combinations

### Miscellaneous

☞• Control place

•▶ Location start of daily work period

◀• Location end of daily work period

🕒➔ From time

➔🕒 To time

🚗➔ From vehicle

OUT➔ Out of scope begin

➔OUT Out of scope end

### Cards

🕒🚗 Driver card

🏢🚗 Company card

☞🚗 Control card














T🚗 Workshop card

🚗--- No card




















### Driving

- ⊗⊗ Crew driving
- ⊗ | Driving time for one week
- ⊗ || Driving time for two weeks








### Printouts

- 24h   Driver activities from card daily printout
- 24h   Driver activities from VU daily printout
- !    Events and faults from card printout
- !    Events and faults from VU printout
- T   Technical data printout
- >>  Over speeding printout

### Events

- !  Insertion of a non valid card
- !   Card conflict
- !   Time overlap
- !   Driving without an appropriate card
- !   Card insertion while driving
- !   Last card session not correctly closed
- >> Over speeding
- !  Power supply interruption
- !   Motion data error
- !   Vehicle motion conflict
- !  Security breach
- !  Time adjustment (by workshop)
- >  Over speeding control

### Faults

- ×  1 Card fault (driver slot)
- ×  2 Card fault (co-driver slot)
- ×  Display fault
- ×  Downloading fault
- ×  Printer fault
- ×  Sensor fault
- ×  VU internal fault

- ×⊠ GNSS fault
- ×⊡ Remote Detection fault

### **Manual entries procedure**

- ⊠?⊠ Still same daily work period ?
- ⊠? End of previous work period ?
- ⊠\*? Confirm or enter location of end of work period
- ⊠⊠? Enter start time
- \*⊠? Enter location of start of work period.

Note: Additional pictogram combinations to form printout blocks or record identifiers are defined in Appendix 4.

## Appendix 4. Printouts

### TABLE OF CONTENT

<b>1. Generalities</b>	<b>305</b>
<b>2. Data blocks specification</b>	<b>305</b>
<b>3. Printout specifications</b>	<b>318</b>
3.1. ....Driver Activities from Card Daily Printout	318
3.2. ....Driver Activities from VU Daily Printout	319
3.3. ....Events and Faults from Card Printout	320
3.4. ....Events and Faults from VU Printout	320
3.5. ....Technical data Printout	321
3.6. ....Over speeding Printout	322
3.7. ....History of inserted cards	323





Block or record number

Data Format
-------------

Meaning

1 ***Date and time at which the document is printed.***

▼ dd/mm/yyyy hh:mm (UTC)
--------------------------

2 ***Type of printout.***

Block identifier

-----▼----- - Picto xxx km/h
------------------------------------

Printout pictogram combination (see App. 3), Speed limiting device setting (Over speeding printout only)

3 ***Card holder identification.***

Block identifier. P= people pictogram

-----P----- -- P Last_Name_____ First_Name_____ Card_Identification_____  dd/mm/yyyy - GEN 2
--

Card holder surname

Card holder first name(s) (if any)

Card identification

Card expiry date (if any) and Card generation number (GEN 1 or GEN 2)\*

In the case where the card is a non-personal card, and holds no card holder surname, the company or workshop or control body name shall be printed instead.

\* The card generation number can only be printed by smart tachograph.

4 ***Vehicle identification.***

Block identifier

-----A----- - A VIN_____ Nat/VRN_____
--

VIN

Registering Member State and VRN

5 ***VU identification.***

Block identifier

-----E----- -
------------------

VU manufacturer's name  
 VU part number  
  
 VU generation number\*

▣ VU_Manufacturer_____ VU_Part_Number__ GEN 2
---

\* The card generation number can only be printed by smart tachograph.

6 **Last calibration of the tachograph**

Block identifier  
  
 Workshop name  
 Workshop card identification  
 Date of the calibration

-----┆----- - ┆ Last_Name_____ Card_Identification_____ ┆ dd/mm/yyyy
--

7 **Last control (by a control officer)**

Block identifier  
  
 Controller's card identification  
 Control date, time and type

-----▣----- - Card_Identification_____ ▣ dd/mm/yyyy hh:mm ppppp
--

Type of the control: Up to five pictograms. The type of control can be (a combination) of:

▣: Card downloading, ▾: VU downloading, ┆: printing, ▣: Displaying, ┆: Roadside calibration checking

8 **Driver activities stored on a card in order of occurrence**

Block identifier  
  
 Enquiry date (calendar day subject of the printout) +  
 Daily card presence counter

-----▣----- -  dd/mm/yyyy xxx
--

8 **Out of scope condition in the beginning of this day**  
 a (leave blank if no out of scope condition open)

-----OUT----- --
---------------------

8. **Period during which the card was not inserted**

1  
 8. Record identifier (start of period)

-----
-------

8.1b	<i>Unknown period.</i> Start time, duration	? hh:mm hhhmm
8.1c	<i>Activity manually entered.</i> Activity pictogram, start time, duration	A hh:mm hhhmm
8.2	<i>Card insertion in slot S</i> Record identifier; S = Slot pictogram Vehicle registering Member State and VRN Vehicle odometer at card insertion	-----S----- A Nat/VRN_____ x xxx xxx km
8.3	<i>Activity (while card was inserted)</i> Activity pictogram, start time, duration, crew status (crew pictogram if CREW, blanks if SINGLE).	A hh:mm hhhmm @@
8.3a	<i>Specific condition.</i> Time of entry, specific condition pictogram (or pictogram combination).	hh:mm ---pppp---
8.4	<i>Card withdrawal</i> Vehicle odometer and distance travelled since last insertion for which odometer is known	x xxx xxx km; x xxx km
9	<i>Driver activities stored in a VU per slot in chronological order</i> Block identifier Enquiry date (calendar day subject of the printout) Vehicle odometer at 00:00 and 24:00	-----@----- - dd/mm/yyyy x xxx xxx - x xxx xxx km
10	<i>Activities carried in slot S</i> Block identifier	-----S----- --
10a	<i>Out of scope condition in the beginning of this day</i> (leave blank if no out of scope condition open)	-----OUT----- --
10.1	<i>Period where no card is inserted in slot S</i> Record identifier. No Card inserted	----- @■---

	Vehicle odometer at beginning of period	x xxx xxx km
1	<i>Card insertion</i>	
0.	Card insertion Record identifier	-----
2	Driver's name	⊙ Last_Name _____
	Driver's first name	First_Name _____
	Driver's Card identification	Card_Identification____ —
	Card expiry date (if any) and Card generation number (GEN 1 or GEN 2)*	dd/mm/yyyy - GEN 2
	Registering MS and VRN of previous vehicle used	A+Nat/VRN _____
	Date and time of card withdrawal from previous vehicle	dd/mm/yyyy hh:mm
	Blank line	
	Vehicle odometer at card insertion, Manual entry of driver activities flag (M if yes, Blank if No).	x xxx xxx km <b>M</b>
	If no card insertion of a driver card happened on the day for which the printout is done then for block 10.2 the odometer data reading from the last available card insertion before that day shall be used.	
1	<i>Activity</i>	
0.	Activity pictogram, start time, duration, crew Status	A hh:mm hh:mm ⊙⊙
3	(crew pictogram if CREW, blanks if SINGLE).	
1	<i>Specific condition.</i> Time of entry, specific condition	hh:mm ---pppp-
0.	pictogram (or pictogram combination).	--
3		
a		
1	<i>Card withdrawal or End of 'No Card' period</i>	
0.	Vehicle odometer at card withdrawal or at end of 'no card' period and distance travelled since insertion, or	x xxx xxx km; x xxx
4	since beginning of the 'No Card' period.	km

\* The card generation number can only be printed by smart tachograph.

1	<i>Daily summary</i>	
1	Block identifier	-----Σ----- -

1	<i>VU summary of periods without card in driver slot</i>	
1.1	Block identifier	1@█---
1	<i>VU summary of periods without card in co-driver slot</i>	
1.2	Block identifier	2@█---
1	<i>VU daily summary per driver</i>	
1.3	Record identifier	-----
	Driver's surname	⊠ Last_Name_____
	Driver's first name(s)	First_Name_____
	Driver's card identification	Card_Identification____
1	<i>Entry of place where a daily work period begins and/or ends</i>	
1.4	pi=location begin / end pictogram, time, country, region,	pihh:mm Cou Reg
	Odometer	x xxx xxx km
1	<i>Entry of place where a daily work period begins and/or ends</i>	
1.5	and after 3 hours continuous driving time	⊠ hh:mm
	Odometer	x xxx xxx km
1	<i>Activity totals (from a card)</i>	
1.6	Total driving duration, distance travelled	⊠ hhhmm x xxx km
	Total working and availability duration	* hhhmm ⊠ hhhmm
	Total resting and unknown duration	⊠ hhhmm ? hhhmm
	Total duration of crew activities	⊠⊠ hhhmm
1	<i>Activity totals (periods without card driver slot)</i>	
1.7	Total driving duration, distance travelled	⊠ hhhmm x xxx km
	Total working and availability duration	* hhhmm ⊠ hhhmm
	Total resting duration	⊠ hhhmm
1	<i>Activity totals (periods without card co-driver slot)</i>	
1.8	Total working and availability duration	* hhhmm ⊠ hhhmm
	Total resting duration	⊠ hhhmm
1	<i>Activity totals (per driver both slots included)</i>	
1.1	Total driving duration, distance travelled	⊠ hhhmm x xxx km

9	Total working and availability duration	* hh <hb>hmm    □ hh<hb>hmm</hb></hb>
	Total resting duration	↳ hh <hb>hmm</hb>
	Total duration of crew activities	⊞⊞ hh <hb>hmm</hb>

When a daily printout is required for the current day, daily summary information is computed with available data at the time of the printout.

1     ***Events and/or faults stored on a card***

2		
1	Block identifier last 5 'Events and Faults' from a card	-----! *□----- -

1	Block identifier all recorded 'Events' on a card	-----! □----- -
---	--	--------------------

1	Block identifier all recorded 'Faults' on a card	-----* □----- -
---	--	--------------------

1     ***Event and/or Fault record***

2.	Record identifier	-----
4	Event/fault pictogram, record purpose, date time of start,	Pic (p) dd/mm/yyyy hh:mm
	Additional event/fault code (if any), duration	!xx hh <hb>hmm</hb>
	Registering Member State & VRN of vehicle in which the event or fault occurred	A Nat/VRN _____

1     ***Events and/or faults stored or on-going in a VU***

3		
1	Block identifier last 5 'Events and Faults' from VU	-----! *A----- -

1	Block identifier all recorded or on-going 'Events' in a VU	-----! A----- -
---	--	--------------------

1	Block identifier all recorded or on-going 'Faults' in a VU	-----* A----- -
---	--	--------------------

1 *Event and/or fault record*

3. Record identifier

4 Event/fault pictogram, record purpose, date time of start,

Additional event/fault code (if any), No of similar events this day, duration

Identification of the cards inserted at start or end of the event or fault (up to 4 lines without repeating twice the same card numbers)

Case where no card was inserted

Manufacturer specific data

```

-----
Pic (p) dd/mm/yyyy hh:mm

!xx (xxx) hhmm

Card_Identification_____
Card_Identification_____
Card_Identification_____
Card_Identification_____

■---

< Literal><ErrorCode>
    
```

The record purpose (p) is a numerical code explaining why the event or fault was recorded, coded in accordance with the data element EventFaultRecordPurpose.

The `Literal` is a tachograph manufacturer specific literal with 12 characters maximum.

The `ErrorCode` is a tachograph manufacturer specific error code with 12 characters maximum.

1 *VU Identification*

4 Block identifier

VU manufacturer name

VU manufacturer address

VU part number

VU approval number

VU serial number

VU year of manufacture

VU software version and installation date

```

-----E-----
-

E
Name_____

Address_____

PartNumber_____

Apprv_____

S/N_____

YYYY

V xxxx dd/mm/yyyy
    
```



1 **Sensor identification**

5 Block identifier

----- -
------------


1 **Pairing record**

5. Sensor serial number  
 1 Sensor approval number  
 1 Sensor pairing date

¶ S/N _____ Apprv _____ dd/mm/yyyy hh:mm
--

1	<b>GNSS identification</b>	
6	Block identifier	

16.1 **Coupling record**

	External GNSS facility serial number	 S/N _____
	External GNSS facility approval number	Apprv _____
	External GNSS facility coupling date	dd/mm/yyyy hh:mm

1 **Calibration data**  
 7  
 Block identifier

-----T-----  
 -

1 **Calibration record**  
 7.  
 1  
 Record identifier  
 Workshop having performed the calibration  
 Workshop address  
 Workshop card identification  
 Workshop card expiry date  
 Blank line  
 Calibration date + calibration purpose  
 VIN  
 Registering Member State & VRN  
 Characteristic coefficient of vehicle  
 Constant of the recording equipment  
 Effective circumference of wheel tyres  
 Size of tyres mounted  
 Speed limiting device setting  
 Old and new odometer values

-----  
 T  
 Workshop\_name \_\_\_\_\_  
  
 Workshop\_address \_\_\_\_\_  
  
 Card\_Identification \_\_\_\_  
 \_\_\_\_  
 dd/mm/yyyy  
  
 T dd/mm/yyyy (p)  
 A VIN \_\_\_\_\_  
  
 Nat/VRN \_\_\_\_\_  
  
 w xx xxx Imp/km  
  
 k xx xxx Imp/km  
  
 l xx xxx mm  
  
 \* TyreSize \_\_\_\_\_  
  
 > xxx km/h  
  
 x xxx xxx - x xxx xxx  
 km

The calibration purpose (p) is a numerical code explaining why these calibration parameters were recorded, coded in accordance with the data element CalibrationPurpose.

1 **Time adjustment**  
 8  
 Block identifier

-----@-----  
 -

1 **Time adjustment record**  
 8.  
 1  
 Record identifier  
 Old date and time  
 New date and time  
 Workshop having performed the time adjustment  
 Workshop address

-----  
 !@ dd/mm/yyyy hh:mm  
  
 @ dd/mm/yyyy hh:mm  
  
 T  
 Workshop\_name \_\_\_\_\_  
  
 Workshop\_address \_\_\_\_\_

Workshop card identification	Card_Identification____
Workshop card expiry date	____ dd/mm/yyyy
<b>1</b> <i>Most recent event and Fault recorded in the VU</i>	
9 Block identifier	-----!x#----- -
Most recent event date time	! dd/mm/yyyy hh:mm
Most recent fault date time	x dd/mm/yyyy hh:mm
<b>2</b> <i>Over speeding control information</i>	
0 Block identifier	----->>----- -
Date and time of last OVER SPEEDING CONTROL	>#dd/mm/yyyy hh:mm
Date/time of first over speeding and number of over speeding events since	>>dd/mm/yyyy hh:mm (nnn)

2 **Over speeding record**

1		
2	Block identifier 'First over speeding after the last calibration'	----->>T-----
1.		
1		
2	Block identifier 'The 5 most serious over the last 365 days'	----->>(365)-----
1.		
2		
2	Block identifier 'The most serious for each of the last 10 days of occurrence'	----->>(10)-----
1.		
3		
2	Record identifier	----- >>dd/mm/yyyy hh:mm hhmm  xxx km/h xxx km/h (xxx)  ☐ Last_Name _____  First_Name _____  Card_Identification____ _____
1.	Date time and duration	
4		
	Max and average speeds, No. of similar events this day	
	Driver's surname	
	Driver's first name(s)	
	Driver card identification	
2	If no over speeding record exists in a block	>>---
1.		
5		

2 **Hand-written information**

2 Block identifier

2 Control Place

2.  
1

2 Controller's signature

2.  
2

2 From time

2.  
3

2 To time

2.  
4

2 Driver's signature

2.  
5

-----  
--  
☐+ .....  
  
☐ .....  
  
☐+ .....  
  
+☐ .....  
  
☐ .....

'Hand-written information'; Insert enough blank lines above a hand-written item, to be able to actually write the required information or to put a signature.

2 **Most recent cards inserted in VU**

3 Block identifier

2 Inserted Card

3.  
1

Record identifier

Type of card, Generation, Version, Manufacturer\*

Card Identification

Card Serial Number

Date and time of last card insertion

----- ☐☐☐ -----  
  
----  
T <gen> <version> <MC>  
Card Identification  
Card Serial Number  
dd/mm/yyyy hh:mm

\* (everything in one line)

with

*type of card*: Pictogram, one character + space

*gen*: GEN1 or GEN2, 4 characters + space

*version*: up to 10 characters

*MC*: manufacturer code, 3 characters

### 3. Printout specifications

In this chapter the following notation conventions have been used:

N
---

Print block or record number N

N
---

Print block or record number N repeated as many times as necessary

X / Y
-------

Print blocks or records X and/or Y as needed, and repeating as many times as necessary.

#### 3.1. Driver Activities from Card Daily Printout

PRT\_008 The driver activities from card daily printout shall be in accordance with the following format:

1	Date and time at which the document is printed
2	Type of printout
3	Controller identification (if a control card is inserted in the VU)
3	Driver identification (from card subject of the printout + GEN)
4	Vehicle identification (vehicle from which printout is taken)
5	VU identification (VU from which printout is taken + GEN)
6	Last calibration of this VU
7	Last control the inspected driver has been subject to
8	Driver activities delimiter
8a	Out of scope condition in the beginning of this day
8.1a / 8.1b / 8.1c / 8.2 / 8.3 / 8.3a / 8.4	Activities of the driver in order of occurrence
11	Daily summary delimiter
11.4	Places entered in chronological order
11.5	GNSS data
11.6	Activity totals
12.1	Events or faults from card delimiter
12.4	Event/Fault records (Last 5 events or faults stored in the card)
13.1	Events or faults from VU delimiter
13.4	Event/Fault records (Last 5 events or faults stored or on-going in the VU)

22.1	Control place
22.2	Controller's signature
22.5	Driver's signature

### 3.2. Driver Activities from VU Daily Printout

PRT\_009 The driver activities from VU daily printout shall be in accordance with the following format:

1	Date and time at which the document is printed
2	Type of printout
3	Card holder identification (for all cards inserted in VU + GEN)
4	Vehicle identification (vehicle from which printout is taken)
5	VU identification (VU from which printout is taken + GEN)
6	Last calibration of this VU
7	Last control on this tachograph
9	Driver activities delimiter
10	Driver slot delimiter (slot 1)
10a	Out of scope condition in the beginning of this day
10.1 / 10.2 / 10.3 /10.3a / 10.4	Activities in chronological order (driver slot)
10	Co-driver slot delimiter (slot 2)
10a	Out of scope condition in the beginning of this day
10.1 / 10.2 / 10.3 /10.3a / 10.4	Activities in chronological order (co-driver slot)
11	Daily summary delimiter
11.1	Summary of periods without card in driver slot
11.4	Places entered in chronological order
11.5	GNSS data
11.6	Activity totals
11.2	Summary of periods without card in co-driver slot
11.4	Places entered in chronological order
11.5	GNSS data
11.7	Activity totals
11.3	Summary of activities for a driver both slots included

11.4	Places entered by this driver in chronological order
11.5	GNSS data
11.8	Activity totals for this driver
13.1	Events faults delimiter
12.4	Event/Fault records (Last 5 events or faults stored or on-going in the VU)
13.1	Control place
22.2	Controller's signature
22.3	From time (space available for a driver without a card to indicate
22.4	To time which periods are relevant to himself)
22.5	Driver's signature

3.3. Events and Faults from Card Printout

PRT\_010 The events and faults from card printout shall be in accordance with the following format:

1	Date and time at which the document is printed
2	Type of printout
3	Controller identification (if a control card is inserted in the VU + GEN)
3	Driver identification (from card subject of the printout)
4	Vehicle identification (vehicle from which printout is taken)
12.2	Events delimiter
12.4	Event records (all events stored on the card)
12.3	Faults delimiter
12.4	Fault records (all faults stored on the card)
22.1	Control place
22.2	Controller's signature
22.5	Driver's signature

3.4. Events and Faults from VU Printout

PRT\_011 The events and faults from VU printout shall be in accordance with the following format:

1	Date and time at which the document is printed
2	Type of printout



3	Card holder identification (for all cards inserted in VU + GEN)
4	Vehicle identification (vehicle from which printout is taken)
13.2	Events delimiter
13.4	Event records (All Events stored or on-going in the VU)
13.3	Faults delimiter
13.4	Fault records (All Faults stored or on-going in the VU)
22.1	Control place
22.2	Controller's signature
22.5	Driver's signature

3.5. Technical data Printout

PRT\_012 The technical data printout shall be in accordance with the following format:

1	Date and time at which the document is printed
2	Type of printout
3	Card holder identification (for all cards inserted in VU + GEN)
4	Vehicle identification (vehicle from which printout is taken)
14	VU identification
15	Sensor identification
15.1	Sensor Pairing data (all data available in chronological order)
16	GNSS identification
16.1	External GNSS facility coupling data (all data available in chronological order)
17	Calibration data delimiter
17.1	Calibration records (all records available in chronological order)
18	Time adjustment delimiter
18.1	Time adjustment records (all records available from time adjustment and from calibration data records)
19	Most recent event and Fault recorded in the VU

3.6. Over speeding Printout

PRT\_013 The over speeding printout shall be in accordance with the following format:

1	Date and time at which the document is printed
2	Type of printout
3	Card holder identification (for all cards inserted in VU + GEN)
4	Vehicle identification (vehicle from which printout is taken)
20	Over speeding control information
21.1	Over speeding data identifier
21.4 / 21.5	First over speeding after the last calibration
21.2	Over speeding data identifier
21.4 / 21.5	The 5 most serious over speeding events over the last 365 days
21.3	Over speeding data identifier
21.4 / 21.5	The most serious over speeding for each of the last 10 days of occurrence
22.1	Control place
22.2	Controller's signature
22.5	Driver's signature

3.7. Historic of inserted cards

PRT\_014 The historic of inserted cards printout shall be in accordance with the following format:

1	Date and time at which the document is printed
2	Type of printout
3	Card holder identifications (of all cards inserted in the V)
23	Most recent card inserted in the VU
23.1	Inserted cards (up to 88 records)
12.3	Faults delimiter

## Appendix 5. Display

In this appendix the following format notation conventions have been used:

- \* characters printed in **bold** denote plain text to be displayed (display remains in normal character),
- \* normal characters denote variables (pictograms or data) to be replaced by their values for displaying:
- \* dd mm yyyy: day, month, year,
- \* hh: hours,
- \* mm: minutes,
- \* D: duration pictogram,
- \* EF: event or fault pictograms combination,
- \* O: mode of operation pictogram.

DIS\_001 The tachograph shall display data using the following formats:

Data	Format
<b>Default display</b>	
Local time	hh:mm
Mode of operation	O
Information related to the driver	<b>1</b> Dhh <b>h</b> mm <b>  </b> hh <b>h</b> mm
Information related to the co-driver	<b>2</b> Dhh <b>h</b> mm
Out of scope condition opened	<b>OUT</b>
<b>Warning display</b>	
Exceeding continuous driving time	<b>1</b> ⊙hh <b>h</b> mm <b>  </b> hh <b>h</b> mm
Event or fault	EF
<b>Other displays</b>	
UTC date time	<b>UTC</b> ⊙dd/mm/yyyy or <b>UTC</b> ⊙dd.mm.yyyy hh:mm
Driver's continuous driving time and cumulative break time	<b>1</b> ⊙hh <b>h</b> mm <b>  </b> hh <b>h</b> mm
Co-driver's continuous driving time and cumulative break time	<b>2</b> ⊙hh <b>h</b> mm <b>  </b> hh <b>h</b> mm
Driver's cumulated driving time for the previous and the current week	<b>1</b> ⊙ <b>  </b> hh <b>h</b> mm

Co-driver's cumulated driving time for the previous and the current week	20   hhhmm
--	------------

## **Appendix 6. Front connector for calibration and download**

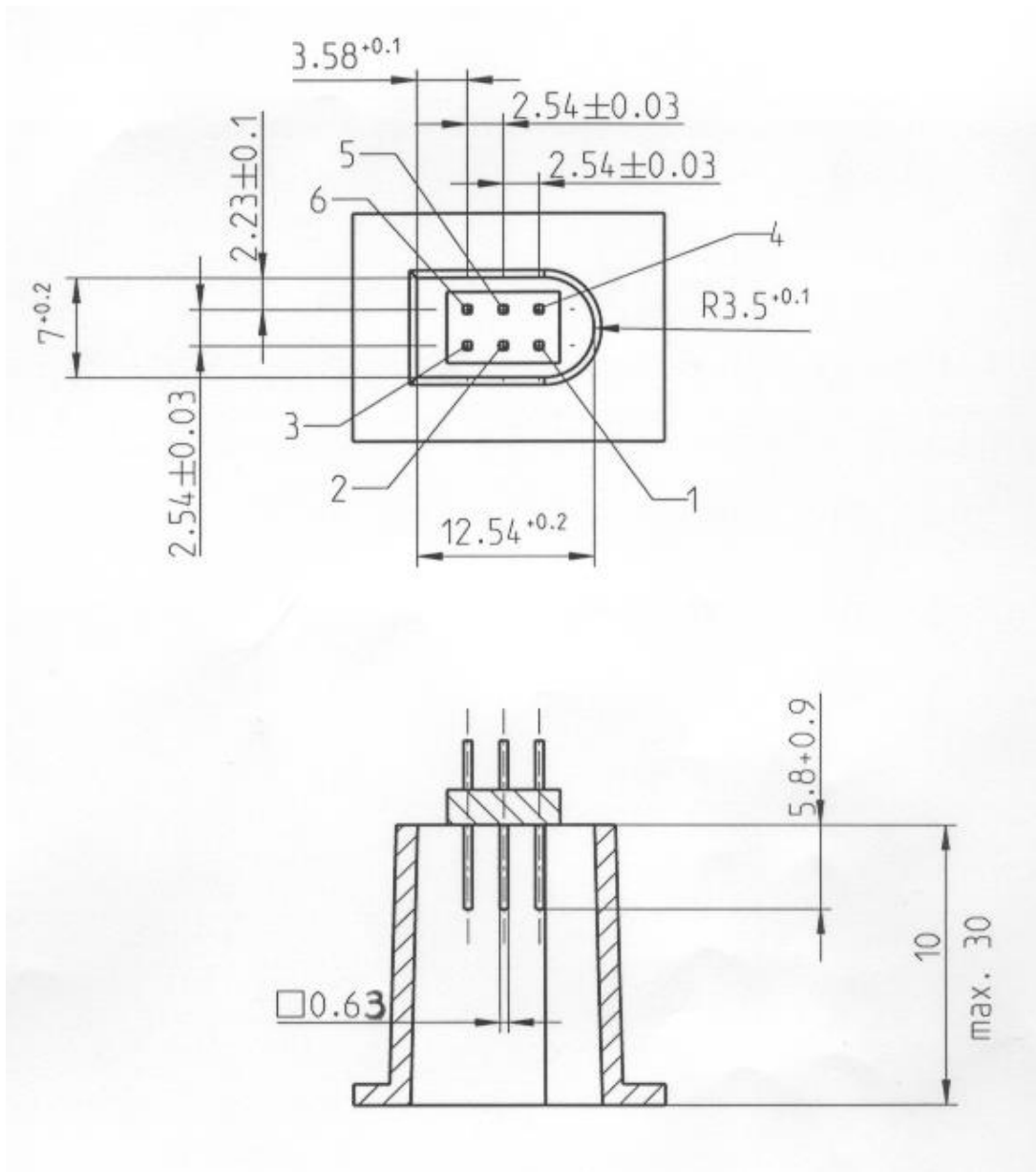
### **TABLE OF CONTENT**

<b>1. Hardware</b>	<b>326</b>
1.1.....Connector	326
1.2.....Contact allocation	328
1.3.....Block diagram	328
<b>2. Downloading interface</b>	<b>328</b>
<b>3. Calibration interface</b>	<b>329</b>

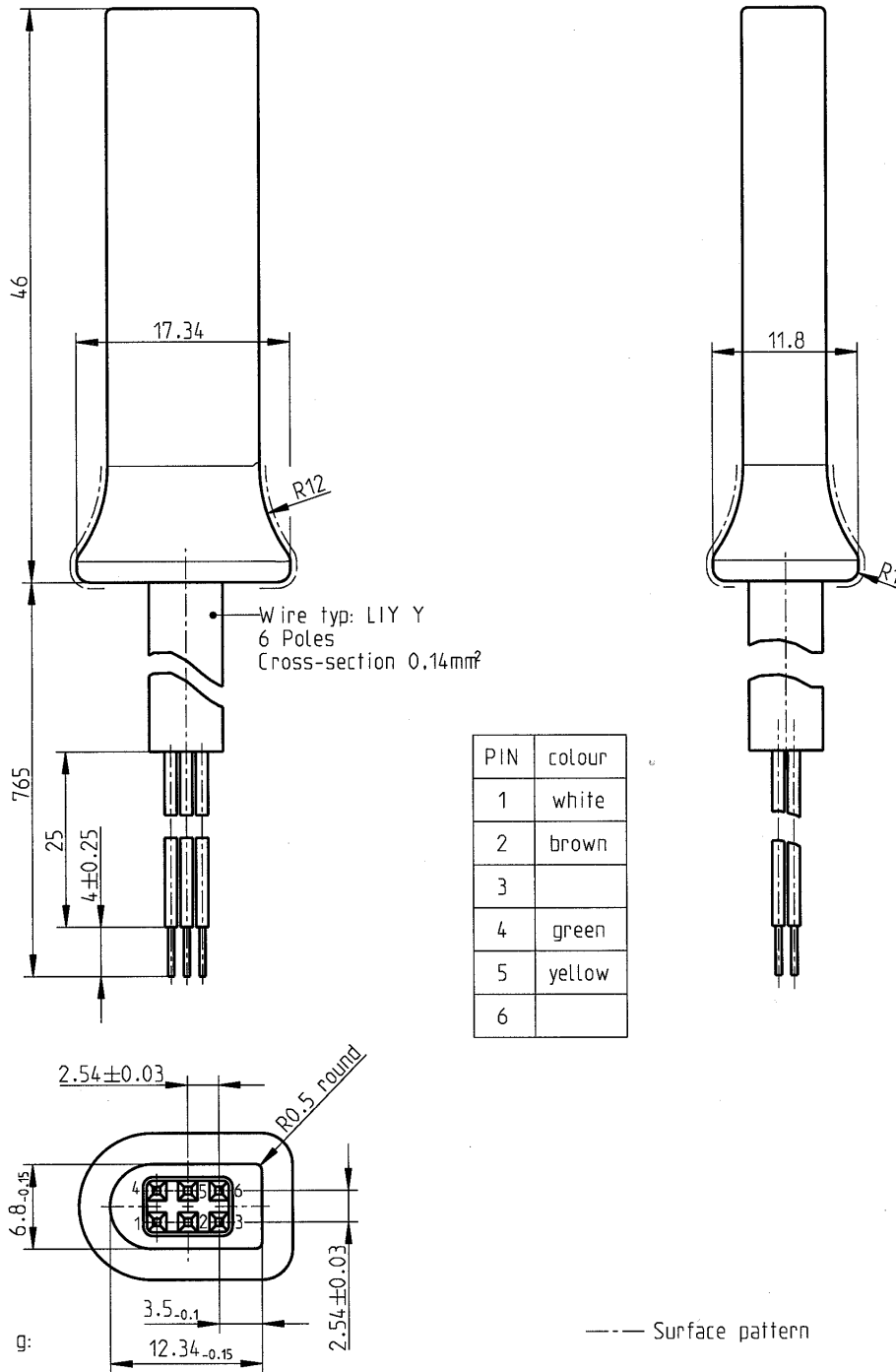
4. Hardware

4.1. Connector

INT\_001 The downloading/calibration connector shall be a 6 pin connector, accessible on the front panel without the need to disconnect any part of the tachograph, and shall comply with the following drawing (all dimensions in millimetres):



The following diagram shows a typical 6 pin mating plug:



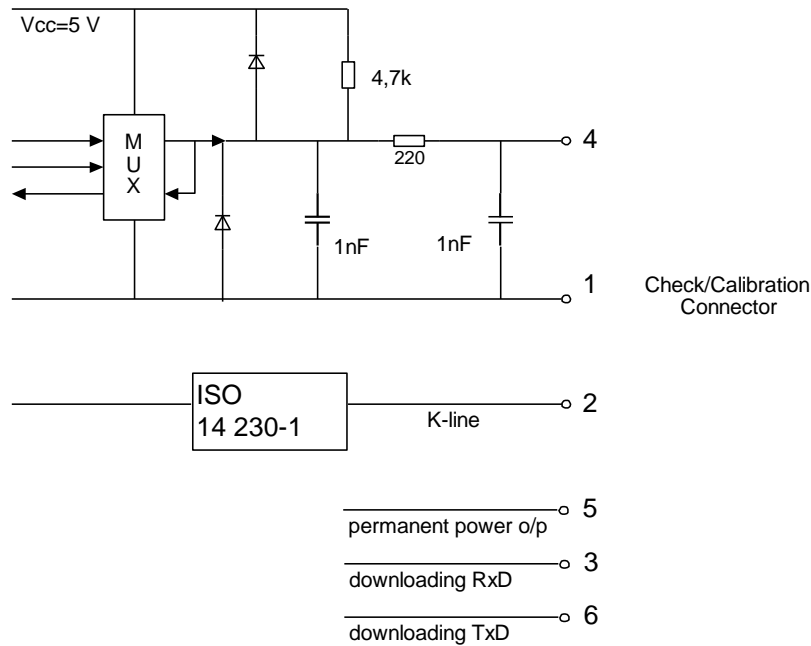
4.2. Contact allocation

INT\_002 Contacts shall be allocated in accordance with the following table:

Pin	Description	Remark
1	Battery minus	Connected to the battery minus of the vehicle
2	Data communication	K-line (ISO 14230-1)
3	RxD – Downloading	Data input to tachograph
4	Input/output signal	Calibration
5	Permanent power output	The voltage range is specified to be that of the vehicle power minus 3V to allow for the voltage drop across the protective circuitry Output 40 mA
6	TxD – Downloading	Data output from tachograph

4.3. Block diagram

INT\_003 The block diagram shall comply with the following:

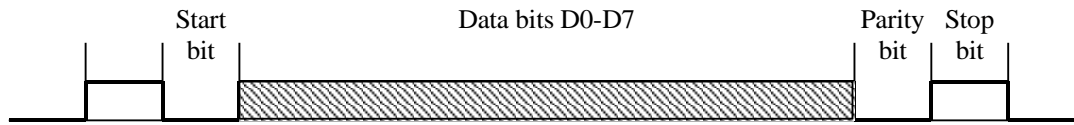


5. Downloading interface

INT\_004 The downloading interface shall comply to RS232 specifications.

INT\_005 The downloading interface shall use one start bit, 8 data bits LSB first, one even parity bit and 1 stop bit.





**Data byte organisation**

- Start bit: one bit with logic level 0;
- Data bits: transmitted with LSB first;
- Parity bit: even parity
- Stop bit: one bit with logic level 1

When numerical data composed by more than one byte are transmitted, the most significant byte is transmitted first and the least significant byte last.

INT\_006 Transmission baud rates shall be adjustable from 9 600 bps to 115 200 bps. Transmission shall be achieved at the highest possible transmission speed, the initial baud rate after a start of communication being set at 9 600 bps.

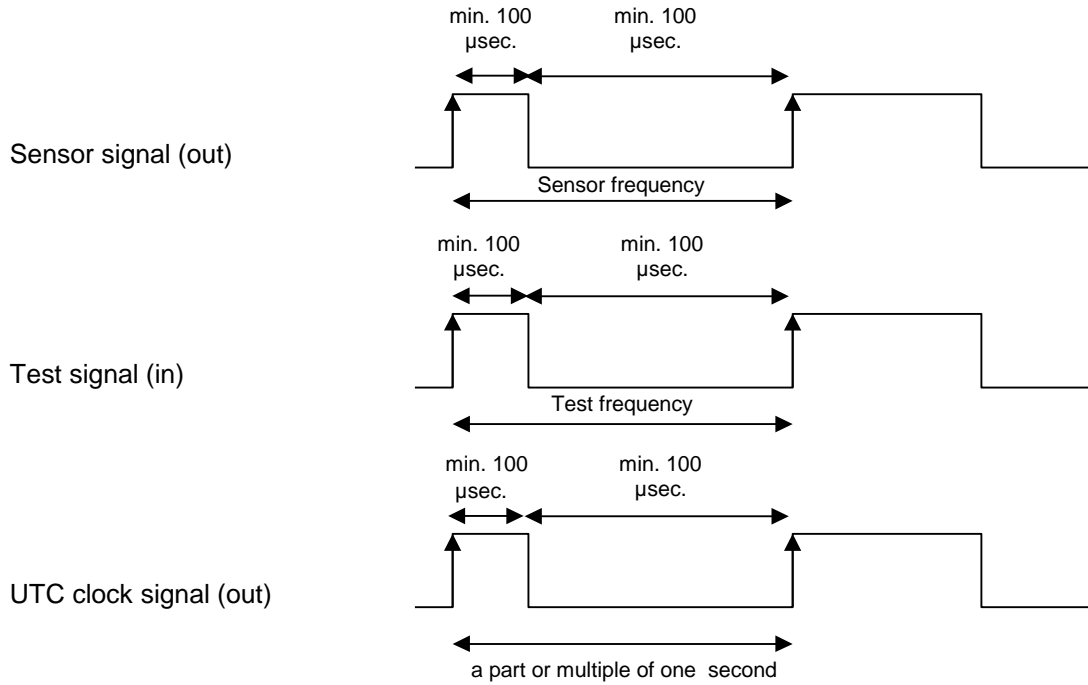
6. Calibration interface

INT\_007 The data communication shall comply to ISO 14230-1 Road vehicles - Diagnostic systems - Keyword protocol 2000 - Part 1: Physical layer, First edition: 1999.

INT\_008 The input/output signal shall comply with the following electrical specification:

Parameter	Minimum	Typical	Maximum	Remark
U <sub>low</sub> (in)			1,0 V	I = 750 μA
U <sub>high</sub> (in)	4 V			I = 200 μA
Frequency			4 kHz	
U <sub>low</sub> (out)			1,0 V	I = 1 mA
U <sub>high</sub> (out)	4 V			I = 1 mA

INT\_009 The input/output signal shall comply with the following timing diagrams:



## Appendix 7. Data downloading protocols

### TABLE OF CONTENT

<b>1. Introduction</b>	<b>333</b>
1.1.....Scope	333
1.2.....Acronyms and notations	333
<b>2. V.U. data downloading</b>	<b>334</b>
2.1.....Download procedure	334
2.2.....Data download protocol	334
2.2.1 Message structure	334
2.2.2 Message types	337
2.2.2.1 Start Communication Request (SID 81)	339
2.2.2.2 Positive Response Start Communication (SID C1)	339
2.2.2.3 Start Diagnostic Session Request (SID 10)	339
2.2.2.4 Positive Response Start Diagnostic (SID 50)	339
2.2.2.5 Link Control Service (SID 87)	339
2.2.2.6 Link Control Positive Response (SID C7)	339
2.2.2.7 Request Upload (SID 35)	339
2.2.2.8 Positive Response Request Upload (SID 75)	340
2.2.2.9 Transfer Data Request (SID 36)	340
2.2.2.10 Positive Response Transfer Data (SID 76)	340
2.2.2.11 Request Transfer Exit (SID 37)	340
2.2.2.12 Positive Response Request Transfer Exit (SID 77)	340
2.2.2.13 Stop Communication Request (SID 82)	340
2.2.2.14 Positive Response Stop Communication (SID C2)	341
2.2.2.15 Acknowledge Sub Message (SID 83)	341
2.2.2.16 Negative Response (SID 7F)	341
2.2.3 Message flow	342
2.2.4 Timing	343
2.2.5 Error handling	343
2.2.5.1 Start Communication phase	344
2.2.5.2 Communication phase	344
2.2.6 Response Message content	347
2.2.6.1 Positive Response Transfer Data Overview	347

2.2.6.2	Positive Response Transfer Data Activities	349
2.2.6.3	Positive Response Transfer Data Events and Faults	352
2.2.6.4	Positive Response Transfer Data Detailed Speed	355
2.2.6.5	Positive Response Transfer Data Technical Data	355
2.3. ....	ESM File storage	358
<b>3.</b>	<b>Tachograph cards downloading protocol</b>	<b>358</b>
3.1. ....	Scope	358
3.2. ....	Definitions	358
3.3. ....	Card Downloading	358
3.3.1	Initialisation sequence	359
3.3.2	Sequence for un-signed data files	359
3.3.3	Sequence for Signed data files	359
3.3.4	Sequence for resetting the calibration counter.	360
3.4. ....	Data storage format	361
3.4.1	Introduction	361
3.4.2	File format	361
<b>4.</b>	<b>Downloading a tachograph card via a vehicle unit.</b>	<b>362</b>

## 1. Introduction

This appendix specifies the procedures to follow in order to perform the different types of data download to an External Storage Medium, together with the protocols that must be implemented to assure the correct data transfer and the full compatibility of the downloaded data format to allow any controller to inspect these data and be able to control their authenticity and their integrity before analysing them.

### 1.1. Scope

Data may be downloaded to an ESM:

- from a Vehicle Unit by an Intelligent Dedicated Equipment (IDE) connected to the VU,
- from a tachograph card by an IDE fitted with a card interface device (IFD),
- from a tachograph card via a vehicle unit by an IDE connected to the VU.

To give the possibility to verify the authenticity and integrity of downloaded data stored on an ESM, data is downloaded with a signature appended in accordance with Appendix 11 Common Security Mechanisms. The source equipment (VU or card) identification and its security certificates (Member state and equipment) are also downloaded. The verifier of the data must possess independently a trusted European public key.

DDP\_001 Data downloaded during one download session must be stored in the ESM within one single file.

### 1.2. Acronyms and notations

The following acronyms are used in this appendix:

**AID** Application Identifier

**ATR** Answer To Reset

**CS** Checksum byte

**DF** Dedicated File

**DS** Diagnostic Session

**EF** Elementary File

**ESM** External Storage Medium

**FID** File Identifier (File ID)

**FMT** Format Byte (first byte of message header)

**ICC** Integrated Circuit Card

**IDE** Intelligent Dedicated Equipment: The equipment used to perform data downloading to the ESM (e.g. Personal Computer)

**IFD** Interface Device

**KWP** Keyword Protocol 2000

**LEN** Length Byte (last byte of message header)

**PPS** Protocol Parameter Selection

**PSO** Perform Security Operation

**SID** Service Identifier

**SRC** Source byte

**TGT** Target Byte

- TLV** Tag Length Value
- TREP** Transfer Response Parameter
- TRTP** Transfer Request Parameter
- VU** Vehicle Unit

2. V.U. data downloading

2.1. Download procedure

In order to carry on a VU data download, the operator must perform the following operations:

- Insert his tachograph card inside a card slot of the VU(\*);
- Connect the IDE to the VU download connector;
- Establish the connection between the IDE and the VU;
- Select on the IDE the data to download and send the request to the VU;
- Close the download session.

(\* The card inserted will trigger the appropriate access rights to the downloading function and to the data. It shall, however, be possible to download data from a driver card inserted into one of the VU slots when no other card type is inserted in the other slot.

2.2. Data download protocol

The protocol is structured on a master-slave basis, with the IDE playing the master role and the VU playing the slave role.

The message structure, types and flow are principally based on the Keyword Protocol 2000 (KWP) (ISO 14230-2 Road vehicles – Diagnostic systems – Keyword protocol 2000 – Part2: Data link layer).

The application layer is principally based on the current draft to date of ISO 14229-1 (Road vehicles – Diagnostic systems – Part 1: Diagnostic services, version 6 of 22 February 2001).

2.2.1 Message structure

DDP\_002 All the messages exchanged between the IDE and the VU are formatted with a structure consisting of three parts:

- Header composed by a Format byte (FMT), a Target byte (TGT), a Source byte (SRC) and possibly a Length byte (LEN),
- Data field composed by a Service Identifier byte (SID) and a variable number of data bytes, which can include an optional diagnostic session byte (DS\_) or an optional transfer parameter byte (TRTP or TREP).
- Checksum composed by a Checksum byte (CS).

Header				Data field					Checksum
FMT	TGT	SRC	LEN	SID	DATA	.....	.....	.....	CS
4 bytes				Max 255 bytes					1 byte

The TGT and SRC byte represent the physical address of the recipient and originator of the message. Values are F0 Hex for the IDE and EE Hex for the VU.

The LEN byte is the length of the Data field part.

The Checksum byte is the 8 bit sum series modulo 256 of all the bytes of the message excluding the CS itself.

FMT, SID, DS\_, TRTP and TREP bytes are defined later in this document.

DDP\_003 In the case where the data to be carried by the message is longer than the space available in the data field part, the message is actually sent in several sub messages. Each sub message bears a header, the same SID, TREP and a 2-byte sub message counter indicating the sub message number within the total message. To enable error checking and abort the IDE acknowledges every sub message. The IDE can accept the sub message, ask for it to be re-transmitted, request the VU to start again or abort the transmission.

DDP\_004 If the last sub message contains exactly 255 bytes in the data field, a final sub message with an empty (except SID TREP and sub message counter) data field must be appended to show the end of the message.

Example:

Header	SI D	TR EP	Message			C S
4 Bytes	Longer than 255 Bytes					

Will be transmitted as:

Header	SI D	TR EP	0 0	01	Sub message 1	C S
4 Bytes	255 Bytes					

Header	SI D	TR EP	0 0	02	Sub message 2	C S
4 Bytes	255 Bytes					

...

Header	SI D	TR EP	x x	yy	Sub message n	C S
4 Bytes	Less than 255 Bytes					

or as:

Header	SI D	TR EP	0 0	01	Sub message 1	C S
4 Bytes	255 Bytes					

Header	SI D	TR EP	0 0	02	Sub message 2	C S
4 Bytes	255 Bytes					

...

Header	SI D	TR EP	x x	yy	Sub message n	C S
4 Bytes	255 Bytes					

Header	SI D	TR EP	x x	yy +1	C S
4 Bytes	4 bytes				



2.2.2 Message types

The communication protocol for data download between the VU and the IDE requires the exchange of 8 different message types.

The following table summarises these messages.

Message Structure

IDE ->	<- VU	Max 4 Bytes Header				Max 255 Bytes Data		1 Byte CheckSum
		F M T	T G T	S R C	L E N	S I D	DS_ / TRTP	DATA
Start Communication Request		81	EE	F0	03	81		E0
Positive Response Start Communication		80	F0	EE	03	81	EA, 8F	9B
Start Diagnostic Session Request		80	EE	F0	02	81		F1
Positive Response Start Diagnostic		80	F0	EE	02	81		31
Link Control Service								
Verify Baud Rate (stage 1)								
9 600 Bd		80	EE	F0	04	87	01,01,01	EC
19 200 Bd		80	EE	F0	04	87	01,01,02	ED
38 400 Bd		80	EE	F0	04	87	01,01,03	EE
57 600 Bd		80	EE	F0	04	87	01,01,04	EF
115 200 Bd		80	EE	F0	04	87	01,01,05	F0
Positive Response Verify Baud Rate		80	F0	EE	02	87	01	28
Transition Baud Rate (stage 2)		80	EE	F0	03	87	02,03	ED
Request Upload		80	EE	F0	0A	85	00,00,00,00,00,FF,FF,FF,FF	99
Positive Response Request Upload		80	F0	EE	03	85	00,FF	D5
Transfer Data Request								
Overview		80	EE	F0	02	86	01	97
Activities		80	EE	F0	06	86	Date	CS

Events & Faults	8 0	E E	F 0	0 2	3 6	03		99
Detailed Speed	8 0	E E	F 0	0 2	3 6	04		9A
Technical Data	8 0	E E	F 0	0 2	3 6	05		9B
Card download	8 0	E E	F 0	0 2	3 6	06	Slot	CS
Positive Response Transfer Data	8 0	F 0	E E	L e n	7 6	TREP	Data	CS
Request Transfer Exit	8 0	E E	F 0	0 1	3 7			96
Positive Response Request Transfer Exit	8 0	F 0	E E	0 1	7 7			D6
Stop Communication Request	8 0	E E	F 0	0 1	8 2			E1
Positive Response Stop Communication	8 0	F 0	E E	0 1	C 2			21
Acknowledge sub message	8 0	E E	F 0	L e n	8 3		Data	CS
Negative responses								
General reject	8 0	F 0	E E	0 3	7 F	Sid Req	10	CS
Service not supported	8 0	F 0	E E	0 3	7 F	Sid Req	11	CS
Sub function not supported	8 0	F 0	E E	0 3	7 F	Sid Req	12	CS
Incorrect Message Length	8 0	F 0	E E	0 3	7 F	Sid Req	13	CS
Conditions not correct or Request sequence error	8 0	F 0	E E	0 3	7 F	Sid Req	22	CS
Request out of range	8 0	F 0	E E	0 3	7 F	Sid Req	31	CS
Upload not accepted	8 0	F 0	E E	0 3	7 F	Sid Req	50	CS
Response pending	8 0	F 0	E E	0 3	7 F	Sid Req	78	CS
Data not available	8 0	F 0	E E	0 3	7 F	Sid Req	FA	CS

## Notes:

- Sid Req = the Sid of the corresponding request.
- TREP = the TRTP of the corresponding request.
- Dark cells denote that nothing is transmitted.
- The term upload (as seen from the IDE) is used for compatibility with ISO 14229. It means the same as download (as seen from the VU).
- Potential 2-byte sub message counters are not shown in this table.
- Slot is the slot number, either "1" (card on driver slot) or "2" (card on co-driver slot)
- In case the slot is not specified, the VU shall select slot 1 if a card is inserted in this slot and it shall select slot 2 only in case it is specifically selected by the user.
- 

## 2.2.2.1 Start Communication Request (SID 81)

DDP\_005 This message is issued by the IDE to establish the communication link with the VU. Initial communications are always performed at 9600 baud (until baud rate is eventually changed using the appropriate Link control services).

## 2.2.2.2 Positive Response Start Communication (SID C1)

DDP\_006 This message is issued by the VU to answer positively to a start communication request. It includes the 2 key bytes 'EA' '8F' indicating that the unit supports protocol with header including target source and length information.

## 2.2.2.3 Start Diagnostic Session Request (SID 10)

DDP\_007 The Start Diagnostic Session request message is issued by the IDE in order to request a new diagnostic session with the VU. The sub function 'default session' (81 Hex) indicates a standard diagnostic session is to be opened.

## 2.2.2.4 Positive Response Start Diagnostic (SID 50)

DDP\_008 The Positive Response Start Diagnostic message is sent by the VU to answer positively to Diagnostic Session Request.

## 2.2.2.5 Link Control Service (SID 87)

DDP\_052 The Link Control Service is used by the IDE to initiate a change in baud rate. This takes place in two steps. In step one the IDE proposes the baud rate change, indicating the new rate. On receipt of a positive message from the VU the IDE sends out confirmation of the baud rate change to the VU (step two). The IDE then changes to the new baud rate. After receipt of the confirmation the VU changes to the new baud rate

## 2.2.2.6 Link Control Positive Response (SID C7)

DDP\_053 The Link Control Positive Response is issued by the VU to answer positively to Link Control Service request (step one). Note that no response is given to the confirmation request (step two).

## 2.2.2.7 Request Upload (SID 35)

DDP\_009 The Request Upload message is issued by the IDE to specify to the VU that a download operation is requested. To meet the requirements of ISO14229 data is included covering address, the size and format details for the data requested. As these are not known to the IDE prior to a download, the memory address is set to 0, format is unencrypted and uncompressed and the memory size is set to the maximum.

#### 2.2.2.8 Positive Response Request Upload (SID 75)

DDP\_010 The Positive Response Request Upload message is sent by the VU to indicate to the IDE that the VU is ready to download data. To meet the requirements of ISO 14229 data is included in this positive response message, indicating to the IDE that further Positive Response Transfer Data messages will include 00FF hex bytes maximum.

#### 2.2.2.9 Transfer Data Request (SID 36)

DDP\_011 The Transfer Data Request is sent by the IDE to specify to the VU the type of data that are to be downloaded. A one byte Transfer Request Parameter (TRTP) indicates the type of transfer.

There are six types of data transfer:

- Overview (TRTP 01),
- Activities of a specified date (TRTP 02),
- Events and faults (TRTP 03),
- Detailed speed (TRTP 04),
- Technical data (TRTP 05),
- Card download (TRTP 06).

DDP\_054 It is mandatory for the IDE to request the overview data transfer (TRTP 01) during a download session as this only will ensure that the VU certificates are recorded within the downloaded file (and allow for verification of digital signature).

In the second case (TRTP 02) the Transfer Data Request message includes the indication of the calendar day (TimeReal format) to be downloaded.

#### 2.2.2.10 Positive Response Transfer Data (SID 76)

DDP\_012 The Positive Response Transfer Data is sent by the VU in response to the Transfer Data Request. The message contains the requested data, with a Transfer Response Parameter (TREP) corresponding to the TRTP of the request.

DDP055 In the first case (TREP 01), the VU will send data helping the IDE operator to choose the data he wants to download further. The information contained within this message is:

- Security certificates,
- Vehicle identification,
- VU current date and time,
- Min and Max downloadable date (VU data),
- Indication of cards presence in the VU,
- Previous download to a company,
- Company locks,
- Previous controls.

#### 2.2.2.11 Request Transfer Exit (SID 37)

DDP\_013 The Request Transfer Exit message is sent by the IDE to inform the VU that the download session is terminated.

#### 2.2.2.12 Positive Response Request Transfer Exit (SID 77)

DDP\_014 The Positive Response Request Transfer Exit message is sent by the VU to acknowledge the Request Transfer Exit.

#### 2.2.2.13 Stop Communication Request (SID 82)

DDP\_015 The Stop Communication Request message is sent by the IDE to disconnect the communication link with the VU.

#### 2.2.2.14 Positive Response Stop Communication (SID C2)

DDP\_016 The Positive Response Stop Communication message is sent by the VU to acknowledge the Stop Communication Request.

#### 2.2.2.15 Acknowledge Sub Message (SID 83)

DDP\_017 The Acknowledge Sub Message is sent by the IDE to confirm receipt of each part of a message that is being transmitted as several sub messages. The data field contains the SID received from the VU and a 2-byte code as follows:

- MsgC +1 Acknowledges correct receipt of sub message number MsgC.  
Request from the IDE to the VU to send next sub message
- MsgC indicates a problem with the receipt of sub message number MsgC.  
Request from the IDE to the VU to send the sub message again.
- FFFF requests termination of the message.

This can be used by the IDE to end the transmission of the VU message for any reason.

The last sub message of a message (LEN byte < 255) may be acknowledged using any of these codes or not acknowledged.

The VU responses that will consist of several sub messages are:

- Positive Response Transfer Data (SID 76)

#### 2.2.2.16 Negative Response (SID 7F)

DDP\_018 The Negative Response message is sent by the VU in response to the above request messages when the VU cannot satisfy the request. The data fields of the message contains the SID of the response (7F), the SID of the request, and a code specifying the reason of the negative response. The following codes are available:

- 10 general reject  
The action cannot be performed for a reason not covered below.
- 11 service not supported  
The SID of the request is not understood.
- 12 sub function not supported  
The DS\_ or TRTP of the request is not understood, or there are no further sub messages to be transmitted.
- 13 incorrect message length  
The length of the received message is wrong.
- 22 conditions not correct or request sequence error  
The required service is not active or the sequence of request messages is not correct.
- 31 Request out of range  
The request parameter record (data field) is not valid.
- 50 upload not accepted  
The request cannot be performed (VU in a non appropriate mode of operation or internal fault of the VU).

- 78 response pending  
The action requested cannot be completed in time and the VU is not ready to accept another request.
- FA data not available  
The data object of a data transfer request are not available in the VU (e.g. no card is inserted, ...).

### 2.2.3 Message flow

A typical message flow during a normal data download procedure is the following:

IDE		VU
Start Communication Request	⇨ ⇩	Positive Response
Start Diagnostic Service Request	⇨ ⇩	Positive Response
Request Upload	⇨ ⇩	Positive Response
Transfer Data Request Overview	⇨ ⇩	Positive Response
Transfer Data Request #2	⇨ ⇩	Positive Response #1
Acknowledge Sub Message #1	⇨ ⇩	Positive Response #2
Acknowledge Sub Message #2	⇨ ⇩	Positive Response #m
Acknowledge Sub Message #m	⇨ ⇩	Positive Response (Data Field<255 Bytes)
Acknowledge Sub Message (optional)	⇨	
...		
Transfer Data Request #n	⇨ ⇩	Positive Response
Request Transfer Exit	⇨ ⇩	Positive Response
Stop Communication Request	⇨ ⇩	Positive Response

2.2.4 Timing

DDP\_019 During normal operation the timing parameters shown in the following figure are relevant:

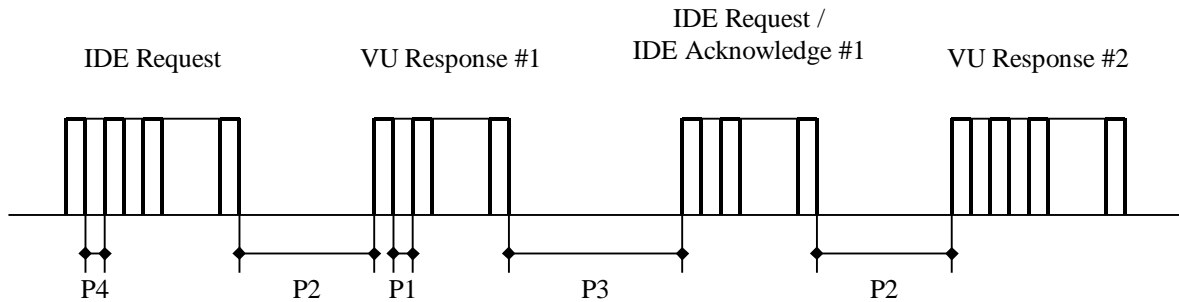


Figure 1 - Message flow, timing

Where:

- P1 = Inter byte time for VU response.
- P2 = Time between end of IDE request and start of VU response, or between end of IDE acknowledge and start of next VU response.
- P3 = Time between end of VU response and start of new IDE request, or between end of VU response and start of IDE acknowledge, or between end of IDE request and start of new IDE request if VU fails to respond.
- P4 = Inter byte time for IDE request.
- P5 = Extended value of P3 for card downloading.

The allowed values for the timing parameters are showed in the following table (KWP extended timing parameters set, used in case of physical addressing for faster communication).

Timing Parameter	Lower limit Value (ms)	Upper limit value (ms)
P1	0	20
P2	20	1000 (*)
P3	10	5000
P4	5	20
P5	10	20 minutes

(\*) if the VU responds with a Negative Response containing a code meaning “request correctly received, response pending”, this value is extended to the same upper limit value of P3.

2.2.5 Error handling

If an error occurs during the message exchange, the message flow scheme is modified depending on which equipment has detected the error and on the message generating the error.

In figure 2 and figure 3 the error handling procedures for the VU and the IDE are respectively shown.

#### 2.2.5.1 Start Communication phase

DDP\_020 If the IDE detects an error during the Start Communication phase, either by timing or by the bit stream, then it will wait for a period P3min before issuing again the request.

DDP\_021 If the VU detects an error in the sequence coming from the IDE, it shall send no response and wait for another Start Communication Request message within a period P3 max.

#### 2.2.5.2 Communication phase

Two different error handling areas can be defined:

##### **1. The VU detects an IDE transmission error.**

DDP\_022 For every received message the VU shall detect timing errors, byte format errors (e.g. start and stop bit violations) and frame errors (wrong number of bytes received, wrong checksum byte).

DDP\_023 If the VU detects one of the above errors, then it sends no response and ignores the message received.

DDP\_024 The VU may detect other errors in the format or content of the received message (e.g. message not supported) even if the message satisfies the length and checksum requirements; in such a case, the VU shall respond to the IDE with a Negative Response message specifying the nature of the error.



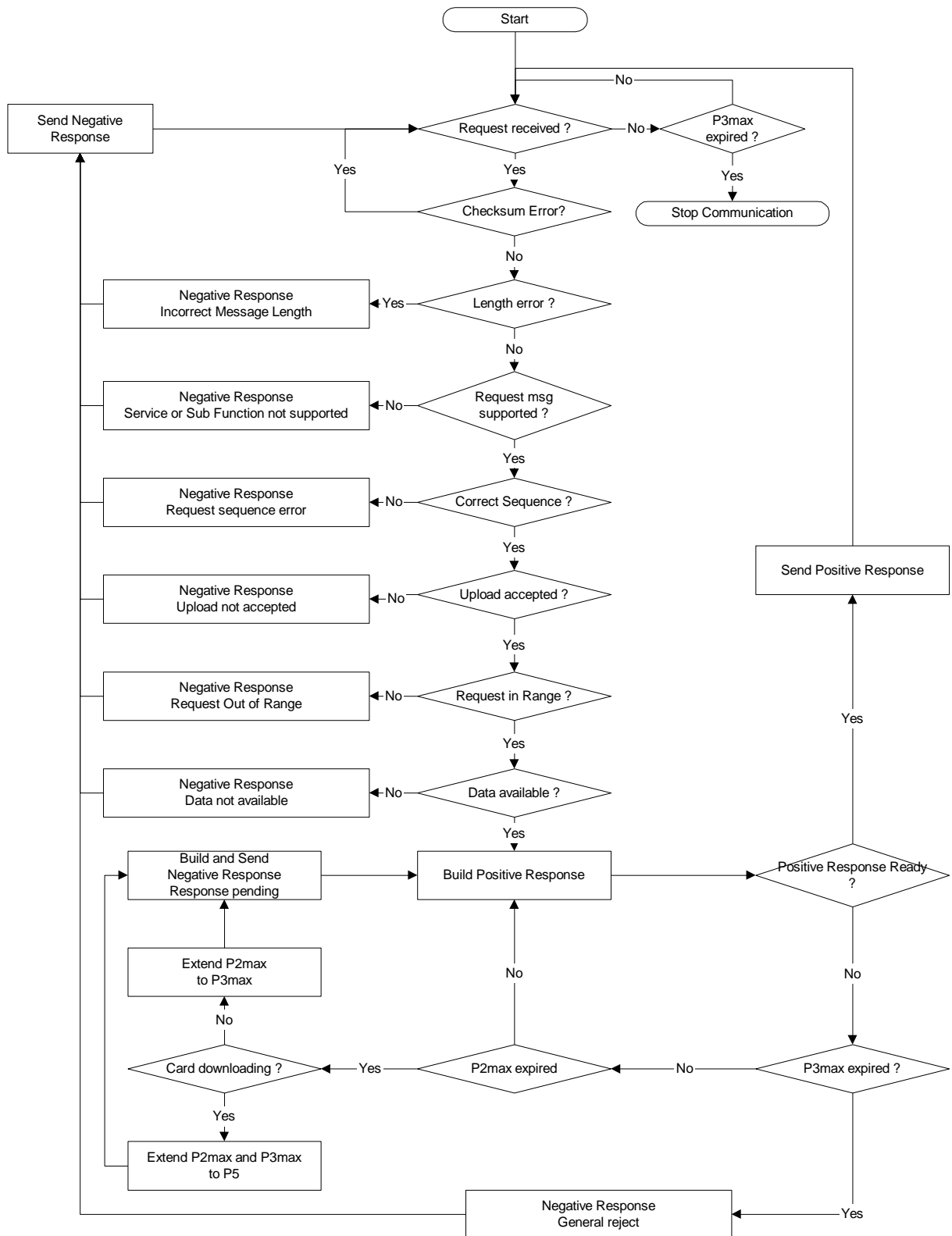


Figure 2 - VU error handling

2. The IDE detects a VU transmission error.

DDP\_025 For every received message the IDE shall detect timing errors, byte format errors (e.g. start and stop bit violations) and frame errors (wrong number of bytes received, wrong checksum byte).

DDP\_026 The IDE shall detect sequence errors, e.g. incorrect sub message counter increments in successive received messages.

DDP\_027 If the IDE detects an error or there was no response from the VU within a P2max period, the request message will be sent again for a maximum of three transmissions in total. For the purposes of this error detection a sub message acknowledge will be considered as a request to the VU.

DDP\_028 The IDE shall wait at least for a period of P3min before beginning each transmission; the wait period shall be measured from the last calculated occurrence of a stop bit after the error was detected.

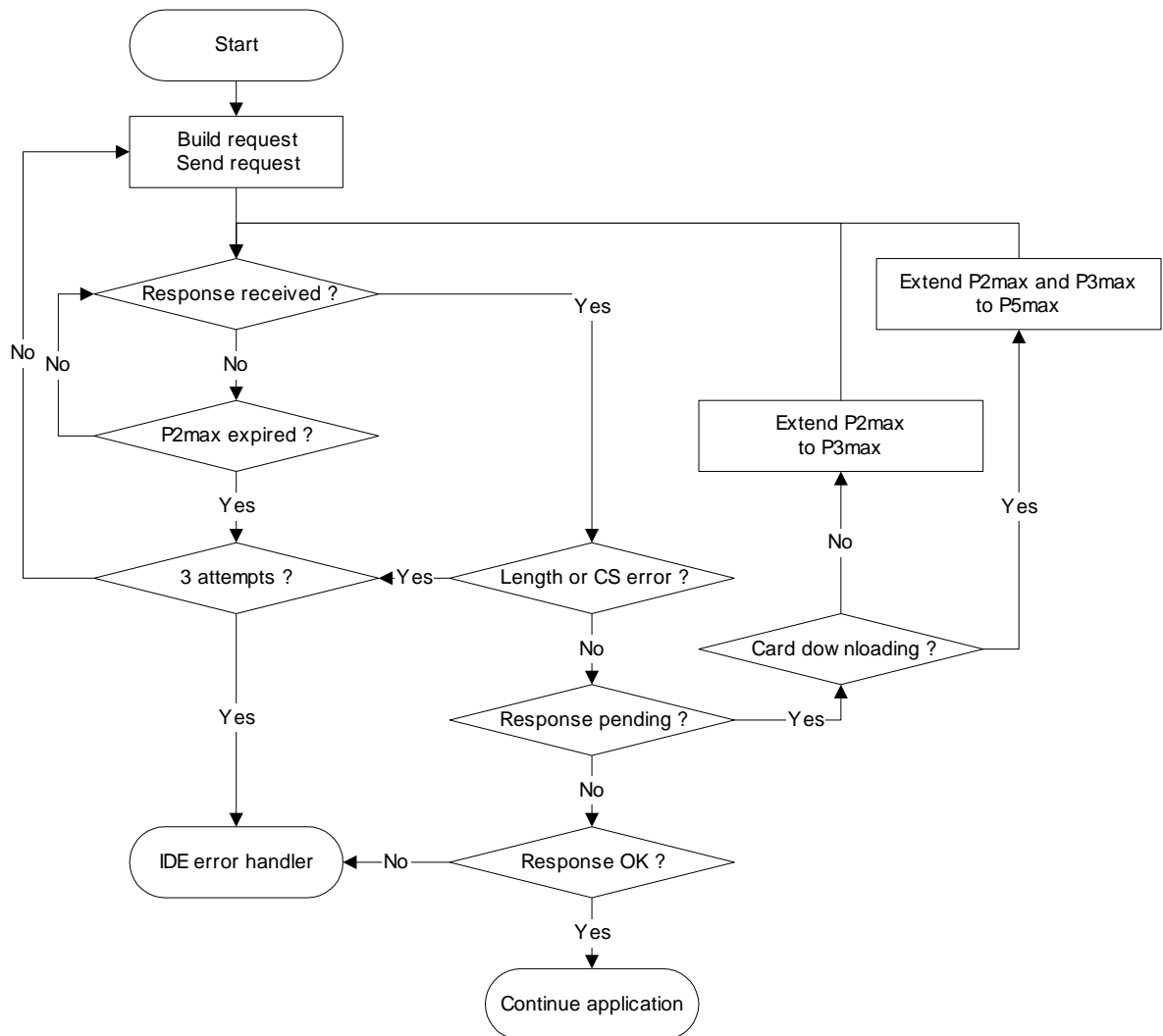


Figure 3 - IDE error handling

## 2.2.6 Response Message content

This paragraph specifies the content of the data fields of the various positive response messages.

Data elements are defined in Appendix 1 data dictionary.

Remark: For generation 2 downloads, each top-level data element is represented by a record array, even if it contains only one record. A record array starts with a header; this header contains the record type, the record size and the number of records. Record arrays are named by "...RecordArray" (with header) in the following tables.

### 2.2.6.1 Positive Response Transfer Data Overview

DDP\_029 The data field of the "Positive Response Transfer Data Overview" message shall provide the following data in the following order under the SID 76 Hex, the TREP 01 Hex and appropriate sub message splitting and counting:

Data structure generation 1

<i>Data element</i>	<i>Comment</i>
MemberStateCertificate	VU Security certificates
VUCertificate	
VehicleIdentificationNumber	Vehicle identification
VehicleRegistrationIdentification	
CurrentDateTime	VU current date and time
VuDownloadablePeriod	Downloadable period
CardSlotsStatus	Type of cards inserted in the VU
VuDownloadActivityData	Previous VU download
VuCompanyLocksData	All company locks stored. If the section is empty, only noOfLocks = 0 is sent.
VuControlActivityData	All control records stored in the VU. If the section is empty, only noOfControls = 0 is sent

<i>Data element</i>	<i>Comment</i>
Signature	RSA signature of all data (except certificates) starting from VehicleIdentificationNumber down to last byte of last VuControlActivityData..
Data structure generation 2	
<b>Data element</b>	<b>Comment</b>
MemberStateCertificateRecordArray	Member state certificate
VUCertificateRecordArray	VU certificate
VehicleIdentificationNumberRecordArray	Vehicle identification
VehicleRegistrationNumberRecordArray	Vehicle registration number
CurrentDateTimeRecordArray	VU current date and time
VuDownloadablePeriodRecordArray	Downloadable period
CardSlotsStatusRecordArray	Type of cards inserted in the VU
VuDownloadActivityDataRecordArray	Previous VU download
VuCompanyLocksRecordArray	All company locks stored. If the section is empty, an array header with noOfRecords = 0 is sent
VuControlActivityRecordArray	All control records stored in the VU. If the section is empty, an array header with noOfRecords = 0 is sent
SignatureRecordArray	ECC signature of all preceding data except the certificates.

2.2.6.2 Positive Response Transfer Data Activities

DDP\_030 The data field of the “Positive Response Transfer Data Activities” message shall provide the following data in the following order under the SID 76 Hex, the TREP 02 Hex and appropriate sub message splitting and counting:

Data structure generation 1

<i>Data element</i>	<i>Comment</i>
TimeReal	Date of day downloaded
OdometerValueMidnight	Odometer at end of downloaded day
VuCardIWData	<p>Cards insertion withdrawal cycles data.</p> <ul style="list-style-type: none"> <li>– If this section contains no available data, only noOfVuCardIWRecords = 0 is sent.</li> <li>– When a VuCardIWRecord lies across 00:00 (card insertion on previous day) or across 24:00 (card withdrawal the following day) it shall appear in full within the two days involved.</li> </ul>
VuActivityDailyData	Slots status at 00:00 and activity changes recorded for the day downloaded.

---

<i>Data element</i>	<i>Comment</i>
VuPlaceDailyWorkPeriodData	Places related data recorded for the day downloaded. If the section is empty, only noOfPlaceRecords = 0 is sent.
VuSpecificConditionData	Specific conditions data recorded for the day downloaded. If the section is empty, only noOfSpecificConditionRecords= 0 is sent
Signature	RSA signature of all data starting from TimeReal down to last byte of last specific condition record.

---

## Data structure generation 2:

---

Data element	Comment
DateOfDayDownloadedRecordArray	Date of day downloaded
OdometerValueMidnightRecordArray	Odometer at end of downloaded day
VuCardIWRecordArray	<p>Cards insertion withdrawal cycles data.</p> <ul style="list-style-type: none"> <li>– If this section contains no available data, an array header with noOfRecords = 0 is sent.</li> <li>– When a VuCardIWRecord lies across 00:00 (card insertion on previous day) or across 24:00 (card withdrawal the following day) it shall appear in full within the two days involved.</li> </ul>
VuActivityDailyRecordArray	Slots status at 00:00 and activity changes recorded for the day downloaded.
VuPlaceDailyWorkPeriodRecordArray	Places related data recorded for the day downloaded. If the section is empty, an array header with noOfRecords = 0 is sent.
VuGNSSCDRecordArray	GNSS positions of the vehicle if the continuous driving time of the driver reaches a multiple of three hours. If the section is empty, an array header with noOfRecords = 0 is sent.
VuSpecificConditionRecordArray	Specific conditions data recorded for the day downloaded. If the section is empty, an array header with noOfRecords = 0 is sent
SignatureRecordArray	ECC signature of all preceding data.

---

2.2.6.3 Positive Response Transfer Data Events and Faults

DDP\_031 The data field of the “Positive Response Transfer Data Events and Faults” message shall provide the following data in the following order under the SID 76 Hex, the TREP 03 Hex and appropriate sub message splitting and counting:

Data structure generation 1

<i>Data element</i>	<i>Comment</i>
VuFaultData	All faults stored or on-going in the VU.  If the section is empty, only noOfVuFaults = 0 is sent.
VuEventData	All events (except over speeding) stored or on-going in the VU.  If the section is empty, only noOfVuEvents = 0 is sent.
VuOverSpeedingControlData	Data related to last over



---

<i>Data element</i>	<i>Comment</i>
	speeding control (default value if no data).
VuOverSpeedingEventData	All over speeding events stored in the VU. If the section is empty, only noOfVuOverSpeedingEvents = 0 is sent.
VuTimeAdjustmentData	All time adjustment events stored in the VU (outside the frame of a full calibration). If the section is empty, only noOfVuTimeAdjRecords = 0 is sent.
Signature	RSA signature of all data starting from noOfVuFaults down to last byte of last time adjustment record

---

---

<i>Data element</i>	<i>Comment</i>
Data structure generation 2:	
<b>Data element</b>	<b>Comment</b>
VuFaultRecordArray	All faults stored or on-going in the VU.  If the section is empty, an array header with noOfRecords = 0 is sent.
VuEventRecordArray	All events (except over speeding) stored or on-going in the VU.  If the section is empty, an array header with noOfRecords = 0 is sent.
VuOverSpeedingControlDataRecordArray	Data related to last over speeding control (default value if no data).
VuOverSpeedingEventRecordArray	All over speeding events stored in the VU.  If the section is empty, an array header with noOfRecords = 0 is sent.
VuTimeAdjustmentRecordArray	All time adjustment events stored in the VU (outside the frame of a full calibration).  If the section is empty, an array header with noOfRecords = 0 is sent.
VuTimeAdjustmentGNSSRecordArray	
SignatureRecordArray	ECC signature of all preceding data.

---

2.2.6.4 Positive Response Transfer Data Detailed Speed

DDP\_032 The data field of the “Positive Response Transfer Data Detailed Speed” message shall provide the following data in the following order under the SID 76 Hex, the TREP 04 Hex and appropriate sub message splitting and counting:

Data structure generation 1

<i>Data element</i>	<i>Comment</i>
VuDetailedSpeedData	All detailed speed stored in the VU (one speed block per minute during which the vehicle has been moving)  60 speed values per minute (one per second).
Signature	RSA signature of all data starting from noOfSpeedBlocks down to last byte of last speed block.

Data structure generation 2:

<b>Data element</b>	<b>Comment</b>
VuDetailedSpeedBlockRecordArray	All detailed speed stored in the VU (one speed block per minute during which the vehicle has been moving)  60 speed values per minute (one per second).
SignatureRecordArray	ECC signature of all preceding data.

2.2.6.5 Positive Response Transfer Data Technical Data

DDP\_033 The data field of the “Positive Response Transfer Data Technical Data” message shall provide the following data in the following order under the SID 76 Hex, the TREP 05 Hex and appropriate sub message splitting and counting:

Data structure generation 1

<i>Data element</i>	<i>Comment</i>
---------------------	----------------

<i>Data element</i>	<i>Comment</i>
VuIdentification	
SensorPaired	
VuCalibrationData	All calibration records stored in the VU.

<i>Data element</i>	<i>Comment</i>
Signature	RSA signature of all data starting from vuManufacturerName down to last byte of last VuCalibrationRecord.

## Data structure generation 2:

<i>Comment</i>	<b>Data element</b>
	VuIdentificationRecordArray
	VuSensorPairedRecordArray
All MS pairings stored in the VU	
	VuSensorExternalGNSSCoupledRecordArray
All external GNSS facility couplings stored in the VU	
	VuCalibrationRecordArray
All calibration records stored in the VU.	
	VuCardRecordArray
All card insertion data stored in the VU.	
	VuITSConsentRecordArray
	VuPowerSupplyInterruptionRecordArray
	SignatureRecordArray
ECC signature of all preceding data.	

### 2.3. ESM File storage

DDP\_034 When a download session has included a VU data transfer, the IDE shall store within one single physical file all data received from the VU during the download session within Positive Response Transfer Data messages. Data stored excludes message headers, sub-message counters, empty sub-messages and checksums but include the SID and TREP (of the first sub-message only if several sub-messages).

## 3. Tachograph cards downloading protocol

### 3.1. Scope

This paragraph describes the direct card data downloading of a tachograph card to an IDE. The IDE is not part of the secure environment; therefore no authentication between the card and the IDE is performed.

### 3.2. Definitions

**Download session:** Each time a download of the ICC data is performed. The session covers the complete procedure from the reset of the ICC by an IFD until the deactivation of the ICC (withdraw of the card or next reset).

**Signed Data File:** A file from the ICC. The file is transferred to the IFD in plain text. On the ICC the file is hashed and signed and the signature is transferred to the IFD.

### 3.3. Card Downloading

DDP\_035 The download of a tachograph card includes the following steps:

- Download the common information of the card in the EFs ICC and IC. This information is optional and is not secured with a digital signature.
- Download the EFs Card\_Certificate (or CardSignCertificate) and CA\_Certificate. This information is not secured with a digital signature.

It is mandatory to download these files for each download session.

- Download the other application data EFs (within Tachograph\_DF and Tachograph\_G2\_DF if relevant) except EF Card\_Download. This information is secured with a digital signature.
- It is mandatory to download at least the EFs Application\_Identification and ID for each download session.
  - When downloading a driver card it is also mandatory to download the following EFs:
    - Events\_Data,
    - Faults\_Data,
    - Driver\_Activity\_Data,
    - Vehicles\_Used,
    - Places,
    - GNSS\_Places (if relevant),
    - Control\_Activity\_Data,
    - Specific\_Conditions.

- When downloading a driver card, update the LastCardDownload date in EF Card\_Download,
- When downloading a workshop card, reset the calibration counter in EF Card\_Download.
- When downloading a workshop card the EF Sensor\_Installation\_Data shall not be downloaded.

3.3.1 Initialisation sequence

DDP\_036 The IDE shall initiate the sequence as follows:

Card	Direction	IDE / IFD	Meaning / Remarks
	↵	Hardware reset	
<b>ATR</b>	⇒		

It is optional to use PPS to switch to a higher baud rate as long as the ICC supports it.

3.3.2 Sequence for un-signed data files

DDP\_037 The sequence to download the EFs ICC, IC, Card\_Certificate (or CardSignCertificate) and CA\_Certificate is as follows:

Card	Direction	IDE / IFD	Meaning / Remarks
	↵	<b>Select File</b>	Select by File identifiers
<b>OK</b>	⇒		
	↵	<b>Read Binary</b>	If the file contains more data than the buffer size of the reader or the card the command has to be repeated until the complete file is read.
<b>File Data OK</b>	⇒	Store data to ESM	according to 3.4 Data storage format

Note 1: Before selecting the Card\_Certificate (or CardSignCertificate) EF, the Tachograph Application must be selected (selection by AID).

Note 2: Selecting and reading a file may also be performed in one step using a Read Binary command with a short EF identifier.

3.3.3 Sequence for Signed data files

DDP\_038 The following sequence shall be used for each of the following files that has to be downloaded with their signature:

Card	Dir	IDE / IFD	Meaning / Remarks
	↵	<b>Select File</b>	
<b>OK</b>	⇒		
	↵	<b>Perform Hash of File</b>	Calculates the hash value over the data content of the selected file using the prescribed hash algorithm in accordance with Appendix 11. This

<i>Card</i>	<i>Dir</i>	<i>IDE / IFD</i>	<i>Meaning / Remarks</i>
			command is not an ISO-Command.
Calculate Hash of File and store Hash value temporarily			
<b>OK</b>	⇒		
	⇐	<b>Read Binary</b>	If the file contains more data than the buffer of the reader or the card can hold, the command has to be repeated until the complete file is read.
<b>File Data</b>	⇒	Store received data to ESM	according to <b>3.4</b> Data storage format
<b>OK</b>	⇐	<b>PSO: Compute Digital Signature</b>	
Perform Security Operation „Compute Digital Signature“ using the temporarily stored Hash value			
<b>Signature</b>	⇒	Append data to the previous stored data on the ESM	according to <b>3.4</b> Data storage format
<b>OK</b>			

Note: Selecting and reading a file may also be performed in one step using a Read Binary command with a short EF identifier. In this case the EF may be selected and read before the command Perform Hash of File is applied.

### 3.3.4 Sequence for resetting the calibration counter.

DDP\_039 The sequence to reset the NoOfCalibrationsSinceDownload counter in the EF Card\_Download in a workshop card is the following:

<i>Card</i>	<i>Dir</i>	<i>IDE / IFD</i>	<i>Meaning / Remarks</i>
	⇐	<b>Select File</b> EF Card_Download	Select by File identifiers
<b>OK</b>	⇒		
	⇐	<b>Update Binary</b> NoOfCalibrationsSinceDownload = '00 00'	
resets card download			



<i>Card</i>	<i>Dir</i>	<i>IDE / IFD</i>	<i>Meaning / Remarks</i>
number			
<b>OK</b>	⇒		

Note: Selecting and updating a file may also be performed in one step using an Update Binary command with a short EF identifier.

### 3.4. Data storage format

#### 3.4.1 Introduction

DDP\_040 The downloaded data has to be stored according to the following conditions:

- The data shall be stored transparent. This means that the order of the bytes as well as the order of the bits inside the byte that are transferred from the card has to be preserved during storage.
- All files of the card downloaded within a download session are stored in one file on the ESM.

#### 3.4.2 File format

DDP\_041 The file format is a concatenation of several TLV objects.

DDP\_042 The tag for an EF shall be the FID plus the appendix „00“.

DDP\_043 The tag of an EF's signature shall be the FID of the file plus the appendix „01“.

DDP\_044 The length is a two byte value. The value defines the number of bytes in the value field. The value „FF FF“ in the length field is reserved for future use.

DDP\_045 When a file is not downloaded nothing related to the file shall be stored (no tag and no zero length).

DDP\_046 A signature shall be stored as the next TLV object directly after the TLV object that contains the data of the file.

<i>Definition</i>	<i>Meaning</i>	<i>Length</i>
FID (2 Bytes)    „00“	Tag for EF (FID)	3 Bytes
FID (2 Bytes)    „01“	Tag for Signature of EF(FID)	3 Bytes
xx xx	Length of Value field	2 Bytes

Example of data in a download file on an ESM:

<i>Tag</i>	<i>Length</i>	<i>Value</i>
00 02 00	00 11	Data of EF ICC
C1 00 00	00 C2	Data of EF Card_Certificate
		...
05 05 00	0A 2E	Data of EF Vehicles_Used
05 05 01	00 80	Signature of EF Vehicles_Used

4. Downloading a tachograph card via a vehicle unit.

DDP\_047 The VU must allow for downloading the content of a driver card inserted to a connected IDE.

DDP\_048 The IDE shall send a “Transfer Data Request Card Download” message to the VU to initiate this mode (see 2.2.2.9).

DDP\_049 The VU shall then download the whole card, file by file, in accordance with the card downloading protocol defined in paragraph 3, and forward all data received from the card to the IDE within the appropriate TLV file format (see 3.4.2) and encapsulated within a “Positive Response Transfer Data” message.

DDP\_050 The IDE shall retrieve card data from the “Positive Response Transfer Data” message (stripping all headers, SIDs, TREPs, sub message counters, and checksums) and store them within one single physical file as described in paragraph 2.3.

DDP\_051 The VU shall then, as applicable, update the `Control_Activity_Data` or the `Card_Download` file of the driver card.

## Appendix 8. Calibration Protocol

### TABLE OF CONTENT

<b>1.</b>	<b>Introduction</b> _____	<b>365</b>
<b>2.</b>	<b>Terms, Definitions and References</b> _____	<b>365</b>
<b>3.</b>	<b>Overview of services</b> _____	<b>366</b>
	3.1. ....Services available	366
	3.2. ....Response codes	366
<b>4.</b>	<b>Communication Services</b> _____	<b>366</b>
	4.1. ....StartCommunication Service	367
	4.2. ....StopCommunication Service	369
	4.2.1    Message description.....	369
	4.2.2    Message format.....	369
	4.2.3    Parameter Definition.....	370
	4.3. ....TesterPresent Service	371
	4.3.1    Message description.....	371
	4.3.2    Message format.....	371
<b>5.</b>	<b>Management Services</b> _____	<b>372</b>
	5.1. ....StartDiagnosticSession service	372
	5.1.1    Message description.....	372
	5.1.2    Message format.....	373
	5.1.3    Parameter definition.....	374
	5.2. ....SecurityAccess service	374
	5.2.1    Message Description.....	375
	5.2.2    Message format - SecurityAccess - requestSeed.....	375
	5.2.3    Message format - SecurityAccess - sendKey.....	377
<b>6.</b>	<b>Data Transmission Services</b> _____	<b>379</b>
	6.1. ....ReadDataByIdentifier service	379
	6.1.1    Message description.....	379
	6.1.2    Message format.....	379
	6.1.3    Parameter Definition.....	380
	6.2. ....WriteDataByIdentifier service	381
	6.2.1    Message description.....	381

6.2.2	Message format.....	381
6.2.3	Parameter definition.....	383
<b>7.</b>	<b>Control of Test Pulses – Input/Output Control functional unit</b> _____	<b>383</b>
7.1. ....	InputOutputControlByIdentifier service	383
7.1.1	Message description.....	383
7.1.2	Message format.....	384
7.1.3	Parameter definition.....	385
<b>8.</b>	<b>dataRecords formats</b> _____	<b>386</b>
8.1. ....	Transmitted parameter ranges	386
8.2. ....	dataRecords formats	387

## 1. Introduction

This appendix describes how data is exchanged between a vehicle unit and a tester via the K-line which forms part of the calibration interface described in Appendix 6. It also describes control of the input / output signal line on the calibration connector.

Establishing K-line communications is described in Section 4 “Communication Services”.

This appendix uses the idea of diagnostic “sessions” to determine the scope of K-line control under different conditions. The default session is the “StandardDiagnosticSession” where all data can be read from a vehicle unit but no data can be written to a vehicle unit.

Selection of the diagnostic session is described in Section 5 “Management Services”.

This appendix has to be considered as relevant for both generations of VUs and of workshop cards, in compliance with the interoperability requirements laid down in this Regulation.

CPR\_001 The “ECUProgrammingSession” allows data entry into the vehicle unit. In the case of entry of calibration data, the vehicle unit must, in addition be in the CALIBRATION mode of operation.

Data transfer via K-line is described in Section 6 “Data Transmission Services”. Formats of data transferred are detailed in Section 8 “dataRecords formats”.

CPR\_002 The “ECUAdjustmentSession” allows the selection of the I/O mode of the calibration I/O signal line via the K-line interface. Control of the calibration I/O signal line is described in section 7 “Control of Test Pulses – Input/Output Control functional unit”.

CPR\_003 Throughout this document the address of the tester is referred to as 'tt'. Although there may be preferred addresses for testers, the VU shall respond correctly to any tester address. The physical address of the VU is 0xEE.

## 2. Terms, Definitions and References

The protocols, messages and error codes are principally based on a draft of ISO 14229-1 (Road vehicles – Diagnostic systems – Part 1: Diagnostic services, version 6 of 22 February 2001).

Byte encoding and hexadecimal values are used for the service identifiers, the service requests and responses, and the standard parameters.

The term ‘tester’ refers to the equipment used to enter programming/calibration data into the VU.

The terms ‘client’ and ‘server’ refer to the tester and the VU respectively.

The term ECU means "Electronic Control Unit" and refers to the VU.

### References:

ISO 14230-2: Road Vehicles -Diagnostic Systems - Keyword Protocol 2000- Part 2 : Data Link Layer. First edition: 1999.

Vehicles – Diagnostic.

3. Overview of services

3.1. Services available

The following table provides an overview of the services that will be available in the tachograph and are defined in this document.

CPR\_004 The table indicates the services that are available in an enabled diagnostic session.

- The **1<sup>st</sup> column** lists the services that are available.
- The **2<sup>nd</sup> column** includes the section number in this appendix where of service is further defined.
- The **3<sup>rd</sup> column** assigns the service identifier values for request messages.
- The **4<sup>th</sup> column** specifies the services of the "**StandardDiagnosticSession**" (**SD**) which must be implemented in each VU.
- The **5<sup>th</sup> column** specifies the services of the "**ECUAdjustmentSession**" (**ECUAS**) which must be implemented to allow control of the I/O signal line in the front panel calibration connector of the VU.
- The **6<sup>th</sup> column** specifies the services of the "**ECUProgrammingSession**" (**ECUPS**) which must be implemented to allow for programming of parameters in the VU.

Diagnostic Service Name	Section No.	SId Req. Value	Diagnostic Sessions		
			SD	ECUAS	ECUPS
StartCommunication	4.1	81	■	■	■
StopCommunication	4.2	82	■		
TesterPresent	4.3	3E	■	■	■
StartDiagnosticSession	5.1	10	■	■	■
SecurityAccess	5.2	27	■	■	■
ReadDataByIdentifier	6.1	22	■	■	■
WriteDataByIdentifier	6.2	2E			■
InputOutputControlByIdentifier	7.1	2F		■	

**Table 1 - Service Identifier value summary table**

- This symbol indicates that the service is mandatory in this diagnostic session.  
No symbol indicates that this service is not allowed in this diagnostic session.

3.2. Response codes

Response codes are defined for each service.

4. Communication Services

Some services are necessary to establish and maintain communication. They do not appear on the application layer. The services available are detailed in the following table:

Service name	Description
StartCommunication	The client requests to start a communication session with a server(s).
StopCommunication	The client requests to stop the current communication session.
TesterPresent	The client indicates to the server that it is still present.

**Table 2 - Communication Services**

CPR\_005 The StartCommunication Service is used for starting a communication. In order to perform any service, communication must be initialised and the communication parameters need to be appropriate for the desired mode.

#### 4.1. StartCommunication Service

CPR\_006 Upon receiving a StartCommunication indication primitive, the VU shall check if the requested communication link can be initialised under the present conditions. Valid conditions for the initialisation of a communication link are described in document ISO 14230-2.

CPR\_007 Then the VU shall perform all actions necessary to initialise the communication link and send a StartCommunication response primitive with the Positive Response parameters selected.

CPR\_008 If a VU that is already initialised (and has entered any diagnostic session) receives a new StartCommunication Request (e.g. due to error recovery in the tester) the request shall be accepted and the VU shall be reinitialised.

CPR\_009 If the communication link cannot be initialised for any reason, the VU shall continue operating as it was immediately prior to the attempt to initialise the communication link..

CPR\_010 The StartCommunication Request message must be physically addressed.

CPR\_011 Initialising the VU for services is performed through a 'fast initialisation' method,

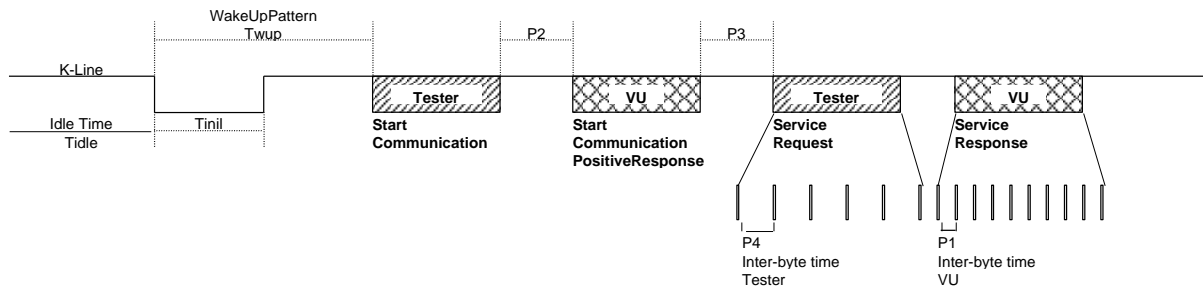
- There is a bus-idle time prior to any activity.
- The tester then sends an initialisation pattern.
- All information which is necessary to establish communication is contained in the response of the VU.

CPR\_012 After completion of the initialisation,

- All communication parameters are set to values defined in Table 4 according to the key bytes.
- The VU is waiting for the first request of the tester.
- The VU is in the default diagnostic mode, i.e. StandardDiagnosticSession.
- The calibration I/O signal line is in the default state, i.e. disabled state.

CPR\_014 The data rate on the K-line shall be 10 400 Baud.

CPR\_016 The fast initialisation is started by the tester transmitting a Wake up pattern (Wup) on the K-line. The pattern begins after the idle time on K-line with a low time of  $T_{inil}$ . The tester transmits the first bit of the StartCommunication Service after a time of  $T_{wup}$  following the first falling edge.



CPR\_017 The timing values for the fast initialisation and communications in general are detailed in the tables below. There are different possibilities for the idle time :

- First transmission after power on, Tidle = 300ms.
- After completion of a StopCommunication Service, Tidle = P3 min.
- After stopping communication by time-out P3 max, Tidle = 0.

Parameter	min value	max value
Tinil	25 ± 1 ms	26 ms
Twup	50 ± 1 ms	51 ms

**Table 3 - Timing values for fast initialisation**

Timing	Parameter Description	lower limit values [ms]	upper limit values [ms]
Parameter	Parameter Description	min.	max.
P1	Inter byte time for VU response	0	20
P2	Time between tester request and VU response or two VU responses	25	250
P3	Time between end of VU responses and start of new tester request	55	5000
P4	Inter byte time for tester request	5	20

**Table 4 - Communication timing values**

CPR\_018 The message format for fast initialisation is detailed in the following tables. (NOTE : Hex means hexadecimal)

Byte #	Parameter Name	Hex Value	Mnemonic
#1	Format byte - physical addressing	81	FMT
#2	Target address byte	EE	TGT
#3	Source address byte	tt	SRC
#4	<b>StartCommunication Request Service Id</b>	<b>81</b>	<b>SCR</b>
#5	Checksum	00-FF	CS



**Table 5 - StartCommunication Request Message**

<i>Byte #</i>	<i>Parameter Name</i>	<i>Hex Value</i>	<i>Mnemonic</i>
#1	Format byte - physical addressing	80	FMT
#2	Target address byte	tt	TGT
#3	Source address byte	EE	SRC
#4	Additional length byte	03	LEN
#5	<b>StartCommunication Positive Response Service Id</b>	<b>C1</b>	<b>SCRPR</b>
#6	Key byte 1	EA	KB1
#7	Key byte 2	8F	KB2
#8	Checksum	00-FF	CS

**Table 6 - StartCommunication Positive Response Message**

CPR\_019 There is no negative response to the StartCommunication Request message, if there is no positive response message to be transmitted then the VU is not initialised, nothing is transmitted and it remains in its normal operation.

#### 4.2. StopCommunication Service

##### 4.2.1 Message description

The purpose of this communication layer service is to terminate a communication session.

CPR\_020 Upon receiving a StopCommunication indication primitive, the VU shall check if the current conditions allow to terminate this communication. In this case the VU shall perform all actions necessary to terminate this communication.

CPR\_021 If it is possible to terminate the communication, the VU shall issue a StopCommunication response primitive with the Positive Response parameters selected, before the communication is terminated.

CPR\_022 If the communication cannot be terminated by any reason, the VU shall issue a StopCommunication response primitive with the Negative Response parameter selected.

CPR\_023 If time-out of P3max is detected by the VU, the communication shall be terminated without any response primitive being issued.

##### 4.2.2 Message format

CPR\_024 The message formats for the StopCommunication primitives are detailed in the following tables.

<i>Byte #</i>	<i>Parameter Name</i>	<i>Hex Value</i>	<i>Mnemonic</i>
#1	Format byte – physical addressing	80	FMT
#2	Target address byte	EE	TGT
#3	Source address byte	tt	SRC

<i>Byte #</i>	<i>Parameter Name</i>	<i>Hex Value</i>	<i>Mnemonic</i>
#4	Additional length byte	01	LEN
<b>#5</b>	<b>StopCommunication Request Service Id</b>	<b>82</b>	<b>SPR</b>
#6	Checksum	00-FF	CS

**Table 7 - StopCommunication Request Message**

<i>Byte #</i>	<i>Parameter Name</i>	<i>Hex Value</i>	<i>Mnemonic</i>
#1	Format byte – physical addressing	80	FMT
#2	Target address byte	tt	TGT
#3	Source address byte	EE	SRC
#4	Additional length byte	01	LEN
<b>#5</b>	<b>StopCommunication Positive Response Service Id</b>	<b>C2</b>	<b>SPRPR</b>
#6	Checksum	00-FF	CS

**Table 8 - StopCommunication Positive Response Message**

<i>Byte #</i>	<i>Parameter Name</i>	<i>Hex Value</i>	<i>Mnemonic</i>
#1	Format byte - physical addressing	80	FMT
#2	Target address byte	tt	TGT
#3	Source address byte	EE	SRC
#4	Additional length byte	03	LEN
<b>#5</b>	<b>negative Response Service Id</b>	<b>7F</b>	<b>NR</b>
#6	StopCommunication Request Service Identification	82	SPR
#7	responseCode = generalReject	10	RC_GR
#8	Checksum	00-FF	CS

**Table 9 - StopCommunication Negative Response Message**

## 4.2.3 Parameter Definition

This service does not require any parameter definition.

### 4.3. TesterPresent Service

#### 4.3.1 Message description

The TesterPresent service is used by the tester to indicate to the server that it is still present, in order to prevent the server from automatically returning to normal operation and possibly stopping the communication. This service, sent periodically, keeps the diagnostic session / communication active by resetting the P3 timer each time a request for this service is received.

#### 4.3.2 Message format

CPR\_079 The message formats for the TesterPresent primitives are detailed in the following tables.

<i>Byte #</i>	<i>Parameter Name</i>	<i>Hex Value</i>	<i>Mnemonic</i>
#1	Format byte – physical addressing	80	FMT
#2	Target address byte	EE	TGT
#3	Source address byte	tt	SRC
#4	Additional length byte	02	LEN
<b>#5</b>	<b>TesterPresent Request Service Id</b>	<b>3E</b>	<b>TP</b>
#6	Sub Function = responseRequired = [ yes no ]	01 02	RESPREQ_ Y RESPREQ_ NO
#7	Checksum	00-FF	CS

**Table 10 - TesterPresent Request Message**

CPR\_080 If the responseRequired parameter is set to 'yes', then the server shall respond with the following positive response message. If set to 'no', then no response is sent by the server.

<i>Byte #</i>	<i>Parameter Name</i>	<i>Hex Value</i>	<i>Mnemonic</i>
#1	Format byte – physical addressing	80	FMT
#2	Target address byte	tt	TGT
#3	Source address byte	EE	SRC
#4	Additional length byte	01	LEN
<b>#5</b>	<b>TesterPresent Positive Response Service Id</b>	<b>7E</b>	<b>TPPR</b>
#6	Checksum	00-FF	CS

**Table 11 - TesterPresent Positive Response Message**

CPR\_081 The service shall support the following negative responses codes:

<i>Byte #</i>	<i>Parameter Name</i>	<i>Hex Value</i>	<i>Mnemonic</i>
#1	Format byte – physical addressing	80	FMT
#2	Target address byte	tt	TGT
#3	Source address byte	EE	SRC
#4	Additional length byte	03	LEN
#5	<b>negative Response Service Id</b>	<b>7F</b>	<b>NR</b>
#6	TesterPresent Request Service Identification	3E	TP
#7	responseCode = [ SubFunctionNotSupported- InvalidFormat incorrectMessageLength ]	12 13	RC_SFNS_I F RC_IML
#8	Checksum	00-FF	CS

**Table 12 - TesterPresent Negative Response Message**

## 5. Management Services

The services available are detailed in the following table:

<i>Service name</i>	<i>Description</i>
StartDiagnosticSession	The client requests to start a diagnostic session with a VU.
SecurityAccess	The client requests access to functions restricted to authorised users.

**Table 13 - Management Services**

### 5.1. StartDiagnosticSession service

#### 5.1.1 Message description

CPR\_025 The service StartDiagnosticSession is used to enable different diagnostic sessions in the server. A diagnostic session enables a specific set of services according to Table 17. A session can enable vehicle manufacturer specific services which are not part of this document. Implementation rules shall conform to the following requirements:

- There shall be always exactly one diagnostic session active in the VU,
- The VU shall always start the StandardDiagnosticSession when powered up. If no other diagnostic session is started, then the StandardDiagnosticSession shall be running as long as the VU is powered,
- If a diagnostic session which is already running has been requested by the tester, then the VU shall send a positive response message,
- Whenever the tester requests a new diagnostic session, the VU shall first send a StartDiagnosticSession positive response message before the new session becomes active in the VU. If the VU is not able to start the requested new diagnostic session, then it shall respond with a StartDiagnosticSession negative response message, and the current session shall continue.

CPR\_026 A diagnostic session shall only be started if communication has been established between the client and the VU.

CPR\_027 The timing parameters defined in Table 4 shall be active after a successful StartDiagnosticSession with the diagnosticSession parameter set to “StandardDiagnosticSession” in the request message if another diagnostic session was previously active.

### 5.1.2 Message format

CPR\_028 The message formats for the StartDiagnosticSession primitives are detailed in the following tables.

<i>Byte #</i>	<i>Parameter Name</i>	<i>Hex Value</i>	<i>Mnemonic</i>
#1	Format byte – physical addressing	80	FMT
#2	Target address byte	EE	TGT
#3	Source address byte	tt	SRC
#4	Additional length byte	02	LEN
<b>#5</b>	<b>StartDiagnosticSession Request Service Id</b>	<b>10</b>	<b>STDS</b>
#6	diagnosticSession = [one value from Table 17]	xx	DS_...
#7	Checksum	00-FF	CS

**Table 14 - StartDiagnosticSession Request Message**

<i>Byte #</i>	<i>Parameter Name</i>	<i>Hex Value</i>	<i>Mnemonic</i>
#1	Format byte – physical addressing	80	FMT
#2	Target address byte	tt	TGT
#3	Source address byte	EE	SRC
#4	Additional length byte	02	LEN
<b>#5</b>	<b>StartDiagnosticSession Positive Response Service Id</b>	<b>50</b>	<b>STDSPR</b>
#6	diagnosticSession = [ same value as in byte #6 Table 14 ]	xx	DS_...
#7	Checksum	00-FF	CS

**Table 15 - StartDiagnosticSession Positive Response Message**

<i>Byte #</i>	<i>Parameter Name</i>	<i>Hex Value</i>	<i>Mnemonic</i>
#1	Format byte – physical addressing	80	FMT
#2	Target address byte	tt	TGT
#3	Source address byte	EE	SRC
#4	Additional length byte	03	LEN
<b>#5</b>	<b>Negative Response Service Id</b>	<b>7F</b>	<b>NR</b>

Byte #	Parameter Name	Hex Value	Mnemonic
#6	StartDiagnosticSession Request Service Id	10	STDS
#7	ResponseCode = [subFunctionNotSupported <sup>a</sup>	12	RC_SFNS
	incorrectMessageLength <sup>b</sup>	13	RC_IML
	conditionsNotCorrect <sup>c</sup>	22	RC_CNC
#8	Checksum	00-FF	CS

**Table 16 - StartDiagnosticSession Negative Response Message**

<sup>a</sup> – the value inserted in byte #6 of the request message is not supported, i.e. not in Table 17,

<sup>b</sup> – the length of the message is wrong,

<sup>c</sup> – the criteria for the request StartDiagnosticSession are not met.

### 5.1.3 Parameter definition

CPR\_029 The parameter *diagnosticSession (DS\_)* is used by the StartDiagnosticSession service to select the specific behaviour of the server(s). The following diagnostic sessions are specified in this document:

Hex	Description	Mnemonic
81	<b>StandardDiagnosticSession</b>  This diagnostic session enables all services specified in <b>Table 1 column 4 "SD"</b> . These services allow reading of data from a server (VU). This diagnostic Session is active after the initialisation has been successfully completed between client (tester) and server (VU). This diagnostic session may be overwritten by other diagnostic sessions specified in this section.	<b>SD</b>
85	<b>ECUProgrammingSession</b>  This diagnostic session enables all services specified in <b>Table 1 column 6 "ECUPS"</b> . These services support the memory programming of a server (VU) This diagnostic session may be overwritten by other diagnostic sessions specified in this section..	<b>ECUPS</b>
87	<b>ECUAdjustmentSession</b>  This diagnostic session enables all services specified in <b>Table 1 column 5 "ECUAS"</b> . These services support the input/output control of a server (VU). This diagnostic session may be overwritten by other diagnostic sessions specified in this section.	<b>ECUAS</b>

**Table 17 - Definition of diagnosticSession Values**

### 5.2. SecurityAccess service

Writing of calibration data is not possible unless the VU is in CALIBRATION mode. In addition to insertion of a valid workshop card into the VU, it is necessary to enter the appropriate PIN into the VU before access to the CALIBRATION mode is granted.

When the VU is in CALIBRATION or CONTROL mode, access to the calibration input/output line is also possible.

The SecurityAccess service provides a means to enter the PIN and to indicate to the tester whether or not the VU is in CALIBRATION mode.

It is acceptable that the PIN may be entered through alternative methods.

### 5.2.1 Message Description

The SecurityAccess service consists of a SecurityAccess "requestSeed" message, eventually followed by a SecurityAccess "sendKey" message. The SecurityAccess service must be carried out after the StartDiagnosticSession service.

- CPR\_033 The tester shall use the SecurityAccess "requestSeed" message to check if the vehicle unit is ready to accept a PIN.
- CPR\_034 If the vehicle unit is already in CALIBRATION mode, it shall answer the request by sending a "seed" of 0x0000 using the service SecurityAccess Positive Response.
- CPR\_035 If the vehicle unit is ready to accept a PIN for verification by a workshop card, it shall answer the request by sending a "seed" greater than 0x0000 using the service SecurityAccess Positive Response.
- CPR\_036 If the vehicle unit is not ready to accept a PIN from the tester, either because the workshop card inserted is not valid, or because no workshop card has been inserted, or because the vehicle unit expects the PIN from another method, it shall answer the request with a Negative Response with a response code set to conditionsNotCorrectOrRequestSequenceError.
- CPR\_037 The tester shall then, eventually, use the SecurityAccess "sendKey" message to forward a PIN to the Vehicle Unit. To allow time for the card authentication process to take place, the VU shall use the negative response code requestCorrectlyReceived-ResponsePending to extend the time to respond. However, the maximum time to respond shall not exceed 5 minutes. As soon as the requested service has been completed, the VU shall send a positive response message or negative response message with a response code different from this one. The negative response code requestCorrectlyReceived-ResponsePending may be repeated by the VU until the requested service is completed and the final response message is sent.
- CPR\_038 The vehicle unit shall answer to this request using the service SecurityAccess Positive Response only when in CALIBRATION mode.
- CPR\_039 In the following cases, the vehicle unit shall answer to this request with a Negative Response with a response code set to:
- subFunctionNot supported : Invalid format for the subfunction parameter (accessType),
  - conditionsNotCorrectOrRequestSequenceError : Vehicle unit not ready to accept a PIN entry,
  - invalidKey : PIN not valid and number of PIN checks attempts not exceeded,
  - exceededNumberOfAttempts : PIN not valid and number of PIN checks attempts exceeded,
  - generalReject : Correct PIN but mutual authentication with workshop card failed.

### 5.2.2 Message format - SecurityAccess - requestSeed

CPR\_040 The message formats for the SecurityAccess "requestSeed" primitives are detailed in the following tables.

<i>Byte #</i>	<i>Parameter Name</i>	<i>Hex Value</i>	<i>Mnemonic</i>
#1	Format byte – physical addressing	80	FMT
#2	Target address byte	EE	TGT
#3	Source address byte	tt	SRC

<i>Byte #</i>	<i>Parameter Name</i>	<i>Hex Value</i>	<i>Mnemonic</i>
#4	Additional length byte	02	LEN
<b>#5</b>	<b>SecurityAccess Request Service Id</b>	<b>27</b>	<b>SA</b>
#6	accessType – requestSeed	7D	AT_RSD
#7	Checksum	00-FF	CS

**Table 18 – SecurityAccess Request- requestSeed Message**

<i>Byte #</i>	<i>Parameter Name</i>	<i>Hex Value</i>	<i>Mnemonic</i>
#1	Format byte – physical addressing	80	FMT
#2	Target address byte	tt	TGT
#3	Source address byte	EE	SRC
#4	Additional length byte	04	LEN
<b>#5</b>	<b>SecurityAccess Positive Response Service Id</b>	<b>67</b>	<b>SAPR</b>
#6	accessType – requestSeed	7D	AT_RSD
#7	Seed High	00-FF	SEEDH
#8	Seed Low	00-FF	SEEDL
#9	Checksum	00-FF	CS

**Table 19 – SecurityAccess - requestSeed Positive Response Message**

<i>Byte #</i>	<i>Parameter Name</i>	<i>Hex Value</i>	<i>Mnemonic</i>
#1	Format byte – physical addressing	80	FMT
#2	Target address byte	tt	TGT
#3	Source address byte	EE	SRC
#4	Additional length byte	03	LEN
<b>#5</b>	<b>negativeResponse Service Id</b>	<b>7F</b>	<b>NR</b>
#6	SecurityAccess Request Service Id	27	SA
#7	responseCode =	22	RC_CNC
	[conditionsNotCorrectOrRequestSequenceError	13	RC_IML
	incorrectMessageLength]		
#8	Checksum	00-FF	CS

**Table 20 – SecurityAccess Negative Response Message**



## 5.2.3 Message format - SecurityAccess - sendKey

CPR\_041 The message formats for the SecurityAccess "sendKey" primitives are detailed in the following tables.

Byte #	Parameter Name	Hex Value	Mnemonic
#1	Format byte – physical addressing	80	FMT
#2	Target address byte	EE	TGT
#3	Source address byte	tt	SRC
#4	Additional length byte	m+2	LEN
<b>#5</b>	<b>SecurityAccess Request Service Id</b>	<b>27</b>	<b>SA</b>
#6	accessType – sendKey	7E	AT_SK
#7 to #m+6	Key #1 (High)	xx	KEY
	...	...	
	Key #m (low, m must be a minimum of 4, and a maximum of 8)	xx	
#m+7	Checksum	00-FF	CS

**Table 21 – SecurityAccess Request - sendKey Message**

Byte #	Parameter Name	Hex Value	Mnemonic
#1	Format byte – physical addressing	80	FMT
#2	Target address byte	tt	TGT
#3	Source address byte	EE	SRC
#4	Additional length byte	02	LEN
<b>#5</b>	<b>SecurityAccess Positive Response Service Id</b>	<b>67</b>	<b>SAPR</b>
#6	accessType – sendKey	7E	AT_SK
#7	Checksum	00-FF	CS

**Table 22 – SecurityAccess - sendKey Positive Response Message**

Byte #	Parameter Name	Hex Value	Mnemonic
#1	Format byte – physical addressing	80	FMT
#2	Target address byte	tt	TGT
#3	Source address byte	EE	SRC
#4	Additional length byte	03	LEN
<b>#5</b>	<b>NegativeResponse Service Id</b>	<b>7F</b>	<b>NR</b>
#6	SecurityAccess Request Service Id	27	SA

<i>Byte #</i>	<i>Parameter Name</i>	<i>Hex Value</i>	<i>Mnemonic</i>
#7	ResponseCode = [generalReject	10	RC_GR
	subFunctionNotSupported	12	RC_SFNS
	incorrectMessageLength	13	RC_IML
	conditionsNotCorrectOrRequestSequenceEr	22	RC_CNC
	ror	35	RC_IK
	invalidKey	36	RC_ENA
	exceededNumberOfAttempts	78	RC_RCR_R
	requestCorrectlyReceived- ResponsePending]		P
#8	Checksum	00-FF	CS

**Table 23 – SecurityAccess Negative Response Message**

## 6. Data Transmission Services

The services available are detailed in the following table:

<i>Service name</i>	<i>Description</i>
ReadDataByIdentifier	The client requests the transmission of the current value of a record with access by recordDataIdentifier.
WriteDataByIdentifier	The client requests to write a record accessed by recordDataIdentifier.

**Table 24 - Data Transmission Services**

### 6.1. ReadDataByIdentifier service

#### 6.1.1 Message description

CPR\_050 The ReadDataByIdentifier service is used by the client to request data record values from a server. The data are identified by a recordDataIdentifier. It is the VU manufacturer's responsibility that the server conditions are met when performing this service.

#### 6.1.2 Message format

CPR\_051 The message formats for the ReadDataByIdentifier primitives are detailed in the following tables.

<i>Byte #</i>	<i>Parameter Name</i>	<i>Hex Value</i>	<i>Mnemonic</i>
#1	Format byte - physical addressing	80	FMT
#2	Target address byte	EE	TGT
#3	Source address byte	tt	SRC
#4	Additional length byte	03	LEN
<b>#5</b>	<b>ReadDataByIdentifier Request Service Id</b>	<b>22</b>	<b>RDBI</b>
#6 to #7	recordDataIdentifier = [a value from Table 28]	xxxx	RDI_...
#8	Checksum	00-FF	CS

**Table 25 - ReadDataByIdentifier Request Message**

<i>Byte #</i>	<i>Parameter Name</i>	<i>Hex Value</i>	<i>Mnemonic</i>
#1	Format byte - physical addressing	80	FMT
#2	Target address byte	tt	TGT
#3	Source address byte	EE	SRC
#4	Additional length byte	m+3	LEN
<b>#5</b>	<b>ReadDataByIdentifier Positive Response Service Id</b>	<b>62</b>	<b>RDBIPR</b>

Byte #	Parameter Name	Hex Value	Mnemonic
#6 and #7	recordDataIdentifier = [the same value as bytes #6 and #7 Table 25]	xxxx	RDI_...
#8 to #m+7	dataRecord[] = [data#1 : data#m]	xx : xx	DREC_D ATA1 : DREC_D TAm
#m+8	Checksum	00-FF	CS

Table 26 - ReadDataByIdentifier Positive Response Message

Byte #	Parameter Name	Hex Value	Mnemonic
#1	Format byte - physical addressing	80	FMT
#2	Target address byte	tt	TGT
#3	Source address byte	EE	SRC
#4	Additional length byte	03	LEN
#5	<b>NegativeResponse Service Id</b>	<b>7F</b>	<b>NR</b>
#6	ReadDataByIdentifier Request Service Id	22	RDBI
#7	ResponseCode= [requestOutOfRange incorrectMessageLength conditionsNotCorrect]	31 13 22	RC_ROOR RC_IML RC_CNC
#8	Checksum	00-FF	CS

Table 27 - ReadDataByIdentifier Negative Response Message

### 6.1.3 Parameter Definition

CPR\_052 The parameter *recordDataIdentifier* (*RDI\_*) in the ReadDataByIdentifier request message identifies a data record.

CPR\_053 recordDataIdentifier values defined by this document are shown in the table below.

The recordDataIdentifier table consists of four columns and multiple lines.

- The **1<sup>st</sup> column (Hex)** includes the “Hex Value” assigned to the recordDataIdentifier specified in the 3<sup>rd</sup> column.
- **The 2<sup>nd</sup> column (Data element)** specifies the data element of Appendix 1 on which the recordDataIdentifier is based (transcoding is sometimes necessary).
- The **3<sup>rd</sup> column (Description)** specifies the corresponding recordDataIdentifier name.
- The **4<sup>th</sup> column (Mnemonic)** specifies the mnemonic of this recordDataIdentifier.

<i>Hex</i>	<i>Data element</i>	<i>recordDataIdentifier Name (see format in Section 8.2)</i>	<i>Mnemonic</i>
F90B	CurrentDateTime	TimeDate	RDI_TD
F912	HighResOdometer	HighResolutionTotalVehicleDistance	RDI_HRTVD
F918	K-ConstantOfRecordingEquipment	Kfactor	RDI_KF
F91C	L-TyreCircumference	LfactorTyreCircumference	RDI_LF
F91D	W-VehicleCharacteristicConstant	WvehicleCharacteristicFactor	RDI_WVCF
F921	TyreSize	TyreSize	RDI_TS
F922	nextCalibrationDate	NextCalibrationDate	RDI_NCD
F92C	SpeedAuthorised	SpeedAuthorised	RDI_SA
F97D	vehicleRegistrationNation	RegisteringMemberState	RDI_RMS
F97E	VehicleRegistrationNumber	VehicleRegistrationNumber	RDI_VRN
F190	VehicleIdentificationNumber	VIN	RDI_VIN

**Table 28 - Definition of recordDataIdentifier values**

CPR\_054 The parameter *dataRecord (DREC\_)* is used by the ReadDataByIdentifier positive response message to provide the data record value identified by the recordDataIdentifier to the client (tester). Data formats are specified in section 8. Additional user optional dataRecords including VU specific input, internal and output data may be implemented, but are not defined in this document.

## 6.2. WriteDataByIdentifier service

### 6.2.1 Message description

CPR\_056 The WriteDataByIdentifier service is used by the client to write data record values to a server. The data are identified by a recordDataIdentifier. It is the VU manufacturer's responsibility that the server conditions are met when performing this service. To update the parameters listed in Table 28 the VU must be in CALIBRATION mode.

### 6.2.2 Message format

CPR\_057 The message formats for the WriteDataByIdentifier primitives are detailed in the following tables.

<i>Byte #</i>	<i>Parameter Name</i>	<i>Hex Value</i>	<i>Mnemonic</i>
#1	Format byte - physical addressing	80	FMT
#2	Target address byte	EE	TGT
#3	Source address byte	tt	SRC
#4	Additional length byte	m+3	LEN

<i>Byte #</i>	<i>Parameter Name</i>	<i>Hex Value</i>	<i>Mnemonic</i>
<b>#5</b>	<b>WriteDataByIdentifier Request Service Id</b>	<b>2E</b>	<b>WDBI</b>
#6 to #7	recordDataIdentifier = [a value from Table 28]	xxxx	RDI_...
#8 to m+7	dataRecord[] = [data#1 : data#m]	xx : xx	DREC_DATA1 : DREC_DATAm
#m+8	Checksum	00-FF	CS

**Table 29 - WriteDataByIdentifier Request Message**

<i>Byte #</i>	<i>Parameter Name</i>	<i>Hex Value</i>	<i>Mnemonic</i>
#1	Format byte - physical addressing	80	FMT
#2	Target address byte	tt	TGT
#3	Source address byte	EE	SRC
#4	Additional length byte	03	LEN
<b>#5</b>	<b>WriteDataByIdentifier Positive Response Service Id</b>	<b>6E</b>	<b>WDBIPR</b>
#6 to #7	recordDataIdentifier = [the same value as bytes #6 and #7 Table 29]	xxxx	RDI_...
#8	Checksum	00-FF	CS

**Table 30 - WriteDataByIdentifier Positive Response Message**

<i>Byte #</i>	<i>Parameter Name</i>	<i>Hex Value</i>	<i>Mnemonic</i>
#1	Format byte – physical addressing	80	FMT
#2	Target address byte	tt	TGT
#3	Source address byte	EE	SRC
#4	Additional length byte	03	LEN
<b>#5</b>	<b>NegativeResponse Service Id</b>	<b>7F</b>	<b>NR</b>
#6	WriteDataByIdentifier Request Service Id	2E	WDBI
#7	ResponseCode= [requestOutOfRange incorrectMessageLength conditionsNotCorrect]	31 13 22	RC_ROOR RC_IML RC_CNC
#8	Checksum	00-FF	CS

**Table 31 - WriteDataByIdentifier Negative Response Message**

## 6.2.3 Parameter definition

The parameter *recordDataIdentifier (RDI\_)* is defined in Table 28.

The parameter *dataRecord (DREC\_)* is used by the WriteDataByIdentifier request message to provide the data record values identified by the recordDataIdentifier to the server (VU). Data formats are specified in section 8.

## 7. Control of Test Pulses – Input/Output Control functional unit

The services available are detailed in the following table:

<i>Service name</i>	<i>Description</i>
InputOutputControlByIdentifier	The client requests the control of an input/output specific to the server.

**Table 32 - Input/Output Control functional unit**

## 7.1. InputOutputControlByIdentifier service

## 7.1.1 Message description

There is a connection via the front connector which allows test pulses to be controlled or monitored using a suitable tester.

CPR\_058 This calibration I/O signal line can be configured by K-line command using the InputOutputControlByIdentifier service to select the required input or output function for the line. The available states of the line are:

- disabled,
- speedSignalInput, where the calibration I/O signal line is used to input a speed signal (test signal) replacing the motion sensor speed signal, this function is not available in CONTROL mode,
- realTimeSpeedSignalOutputSensor, where the calibration I/O signal line is used to output the speed signal of the motion sensor,
- RTCOutput, where the calibration I/O signal line is used to output the UTC clock signal, this function is not available in CONTROL mode.

CPR\_059 The vehicle unit must have entered an adjustment session and must be in CALIBRATION or CONTROL mode to configure the state of the line. When the VU is in CALIBRATION mode, the four states of the line can be selected (disabled, speedSignalInput, realTimeSpeedSignalOutputSensor, RTCOutput). When the VU is in CONTROL mode, only two states of the lines can be selected (disabled, realTimeSpeedOutputSensor). On exit of the adjustment session or of the CALIBRATION or CONTROL mode the vehicle unit must ensure the calibration I/O signal line is returned to the 'disabled' (default) state.

CPR\_060 If speed pulses are received at the real time speed signal input line of the VU while the calibration I/O signal line is set to input then the calibration I/O signal line shall be set to output or returned to the disabled state.

CPR\_061 The sequence shall be:

- Establish communications by StartCommunication Service
- Enter an adjustment session by StartDiagnosticSession Service and be in CALIBRATION or CONTROL mode of operation (the order of these two operation is not important).
- Change the state of the output by InputOutputControlByIdentifier Service.

#### 7.1.2 Message format

CPR\_062 The message formats for the InputOutputControlByIdentifier primitives are detailed in the following tables.

<i>Byte #</i>	<i>Parameter Name</i>	<i>Hex Value</i>	<i>Mnemonic</i>
#1	Format byte - physical addressing	80	FMT
#2	Target address byte	EE	TGT
#3	Source address byte	tt	SRC
#4	Additional length byte	xx	LEN
<b>#5</b>	<b>InputOutputControlByIdentifier Request Sid</b>	<b>2F</b>	<b>IOCBI</b>
#6 and #7	InputOutputIdentifier = [CalibrationInputOutput]	F960	IOI_CIO
#8 or	ControlOptionRecord = [		COR_...
#8 to #9	inputOutputControlParameter - one value from Table 36	xx	IOCP_...
	controlState – one value from Table 37 (see note below)]	xx	CS_...
#9 or #10	Checksum	00-FF	CS

**Table 33 - InputOutputControlByIdentifier Request Message**

**Note:** The controlState parameter is present only in some cases (see 7.1.3).

<i>Byte #</i>	<i>Parameter Name</i>	<i>Hex Value</i>	<i>Mnemonic</i>
#1	Format byte – physical addressing	80	FMT
#2	Target address byte	tt	TGT
#3	Source address byte	EE	SRC
#4	Additional length byte	xx	LEN
<b>#5</b>	<b>inputOutputControlByIdentifier Positive Response SId</b>	<b>6F</b>	<b>IOCBIPR</b>
#6 and #7	inputOutputIdentifier = [CalibrationInputOutput]	F960	IOI_CIO



Byte #	Parameter Name	Hex Value	Mnemonic
#8 or	controlStatusRecord = [		CSR_
#8 to #9	inputOutputControlParameter (same value as byte #8 Table 33)	xx	IOCP_...
	controlState (same value as byte #9 Table 33)] (if applicable)	xx	CS_...
#9 or #10	Checksum	00-FF	CS

**Table 34 - InputOutputControlByIdentifier Positive Response Message**

Byte #	Parameter Name	Hex Value	Mnemonic
#1	Format byte – physical addressing	80	FMT
#2	Target address byte	tt	TGT
#3	Source address byte	EE	SRC
#4	Additional length byte	03	LEN
<b>#5</b>	<b>negativeResponse Service Id</b>	<b>7F</b>	<b>NR</b>
#6	inputOutputControlByIdentifier Request SId	2F	IOCBI
#7	responseCode=[		
	incorrectMessageLength	13	RC_IML
	conditionsNotCorrect	22	RC_CNC
	requestOutOfRange	31	RC_ROOR
	deviceControlLimitsExceeded]	7A	RC_DCLE
#8	Checksum	00-FF	CS

**Table 35 - InputOutputControlByIdentifier Negative Response Message**

### 7.1.3 Parameter definition

CPR\_064 The parameter *inputOutputControlParameter (IOCP\_)* is defined in the following table.

Hex	Description	Mnemonic
00	<b>ReturnControlToECU</b>	RCTECU
	This value shall indicate to the server (VU) that the tester does no longer have control about the calibration I/O signal line.	
01	<b>ResetToDefault</b>	RTD
	This value shall indicate to the server (VU) that it is requested to reset the calibration I/O signal line to its default state.	

<i>Hex</i>	<i>Description</i>	<i>Mnemonic</i>
03	<b>ShortTermAdjustment</b>  This value shall indicate to the server (VU) that it is requested to adjust the calibration I/O signal line to the value included in the controlState parameter.	STA

**Table 36 - Definition of inputOutputControlParameter values**

CPR\_065 The parameter *controlState* is present only when the inputOutputControlParameter is set to ShortTermAdjustment and is defined in the following table:

<i>Mode</i>	<i>Hex Value</i>	<i>Description</i>
Disable	00	I/O line is disabled (default state)
Enable	01	Enable calibration I/O line as speedSignalInput
Enable	02	Enable calibration I/O line as realTimeSpeedSignalOutputSensor
Enable	03	Enable calibration I/O line as RTCOutput

**Table 37 - Definition of controlState values**

#### 8. dataRecords formats

This section details:

- general rules that shall be applied to ranges of parameters transmitted by the vehicle unit to the tester,
- formats that shall be used for data transferred via the Data Transmission Services described in section 6.

CPR\_067 All parameters identified shall be supported by the VU.

CPR\_068 Data transmitted by the VU to the tester in response to a request message shall be of the measured type (i.e. current value of the requested parameter as measured or observed by the VU).

#### 8.1. Transmitted parameter ranges

CPR\_069 Table 38 defines the ranges used to determine the validity of a transmitted parameter.

CPR\_070 The values in the range «error indicator» provide a means for the vehicle unit to immediately indicate that valid parametric data is not currently available due to some type of error in the tachograph.

CPR\_071 The values in the range «not available» provide a means for the vehicle unit to transmit a message which contains a parameter that is not available or not supported in that module. The values in the range «not requested» provide a means for a device to transmit a command message and identify those parameters where no response is expected from the receiving device.

CPR\_072 If a component failure prevents the transmission of valid data for a parameter, the error indicator as described in Table 38 should be used in place of that parameter's data. However, if the measured or calculated data has yielded a value that is valid yet exceeds the defined parameter range, the error indicator should not be used. The data should be transmitted using the appropriate minimum or maximum parameter value.

<i>Range Name</i>	<i>1 byte (Hex value)</i>	<i>2 bytes (Hex value)</i>	<i>4 bytes (Hex Value)</i>	<i>ASCII</i>
Valid signal	00 to FA	0000 to FAFF	00000000 to FAFFFFFF	1 to 254
Parameter specific indicator	FB	FB00 to FBFF	FB000000 to FBFFFFFF	none
Reserved range for future indicator bits	FC to FD	FC00 to FDFF	FC000000 to FDFFFFFF	none
Error indicator	FE	FE00 to FEFF	FE000000 to FEFFFFFF	0
Not available or not requested	FF	FF00 to FFFF	FF000000 to FFFFFF	FF

**Table 38 – dataRecords ranges**

CPR\_073 For parameters coded in ASCII, the ASCII character “\*” is reserved as a delimiter.

## 8.2. dataRecords formats

Table 39 to Table 42 below detail the formats that shall be used via the ReadDataByIdentifier and WriteDataByIdentifier Services.

CPR\_074 Table 39 provides the length, resolution and operating range for each parameter identified by its recordDataIdentifier:

<i>Parameter Name</i>	<i>Data length (bytes)</i>	<i>Resolution</i>	<i>Operating range</i>
TimeDate	8	See details in Table 40	
HighResolutionTotalVehicle Distance	4	5 m/bit gain, 0 m offset	0 to +21 055 406 km
Kfactor	2	0.001 pulse/m /bit gain, 0 offset	0 to 64.255 pulse/m
LfactorTyreCircumference	2	0.125 10 <sup>-3</sup> m /bit gain, 0 offset	0 to 8.031 m
WvehicleCharacteristicFacto r	2	0.001 pulse/m /bit gain, 0 offset	0 to 64.255 pulse/m
TyreSize	15	ASCII	ASCII
NextCalibrationDate	3	See details in Table 41	
SpeedAuthorised	2	1/256 km/h/bit gain, 0 offset	0 to 250.996 km/h
RegisteringMemberState	3	ASCII	ASCII
VehicleRegistrationNumber	14	See details in Table 42	

<i>Parameter Name</i>	<i>Data length (bytes)</i>	<i>Resolution</i>	<i>Operating range</i>
VIN	17	ASCII	ASCII

**Table 39 – Format of dataRecords**

CPR\_075 Table 40 details the formats of the different bytes of the TimeDate parameter :

<i>Byte</i>	<i>Parameter definition</i>	<i>Resolution</i>	<i>Operating range</i>
1	Seconds	0.25 s/bit gain, 0 s offset	0 to 59.75s
2	Minutes	1 min/bit gain, 0 min offset	0 to 59 min
3	Hours	1 h/bit gain, 0 h offset	0 to 23 h
4	Month	1 month/bit gain, 0 month offset	1 to 12 month
5	Day	0.25 day/bit gain, 0 day offset (see NOTE below Table 41)	0.25 to 31.75 day
6	Year	1 year/bit gain, +1985 year offset (see NOTE below Table 41)	1985 to 2235 year
7	Local Minute Offset	1 min/bit gain, -125 min offset	-59 to +59 min
8	Local Hour Offset	1 h/bit gain, -125 h offset	- 23 to +23 h

**Table 40 - Detailed format of TimeDate (recordDataIdentifier value # F90B)**

CPR\_076 Table 41 details the formats of the different bytes of the NextCalibrationDate parameter.

<i>Byte</i>	<i>Parameter definition</i>	<i>Resolution</i>	<i>Operating range</i>
1	Month	1 month/bit gain, 0 month offset	1 to 12 month
2	Day	0.25 day/bit gain, 0 day offset (see NOTE below)	0.25 to 31.75 day
3	Year	1 year/bit gain, +1985 year offset (see NOTE below)	1985 to 2235 year

**Table 41 - Detailed format of NextCalibrationDate (recordDataIdentifier value # F922)**

NOTE concerning the use of the “Day” parameter:

- 1) A value of 0 for the date is null. The values 1, 2, 3, and 4 are used to identify the first day of the month; 5, 6, 7, and 8 identify the second day of the month; etc.
- 2) This parameter does not influence or change the hours parameter above.

NOTE concerning the use of byte “Year” parameter:

A value of 0 for the year identifies the year 1985; a value of 1 identifies 1986; etc.

CPR\_078 Table 42 details the formats of the different bytes of the VehicleRegistrationNumber parameter:

<i>Byte</i>	<i>Parameter definition</i>	<i>Resolution</i>	<i>Operating range</i>
1	Code Page (as defined in Appendix 1)	ASCII	01 to 0A
2 – 14	Vehicle Registration Number (as defined in Appendix 1)	ASCII	ASCII

**Table 42 - Detailed format of VehicleRegistrationNumber (recordDataIdentifier value # F97E)**

## **Appendix 9. Type approval List of minimum required tests**

### **TABLE OF CONTENT**

<b>1. INTRODUCTION</b>	<b>391</b>
<b>2. VEHICLE UNIT FUNCTIONAL TESTS</b>	<b>392</b>
<b>3. MOTION SENSOR FUNCTIONAL TESTS</b>	<b>398</b>
<b>4. TACHOGRAPH CARDS FUNCTIONAL TESTS</b>	<b>402</b>
<b>5. EXTERNAL GNSS FACILITY TESTS</b>	<b>411</b>
<b>6. REMOTE COMMUNICATION FACILITY TESTS</b>	<b>415</b>
<b>7. PAPER FUNCTIONAL TESTS</b>	<b>417</b>
<b>8. INTEROPERABILITY TESTS</b>	<b>419</b>

## 1. Introduction

Type approval The EC type approval for a recording equipment (or component) or a tachograph card is based on:

- a **security certification**, based on Common Criteria specifications, against a security target fully compliant with Appendix 10 to this Annex (To be completed/modified),
- a **functional certification** performed by a Member State authority certifying that the item tested fulfils the requirements of this Annex in terms of functions performed, measurement accuracy and environmental characteristics,
- an **interoperability certification** performed by the competent body certifying that the recording equipment (or tachograph card) is fully interoperable with the necessary tachograph card (or recording equipment) models (see Chapter 8 of this Annex).

This Appendix specifies which tests, as a minimum, must be performed by a Member State authority during the functional tests, and which tests, as a minimum, must be performed by the competent body during the interoperability tests. Procedures to follow to carry out the tests or the type of tests are not specified further.

The security certification aspects are not covered by this Appendix. If some tests requested for type approval are performed during the security evaluation and certification process, then these tests do not need to be performed again. In this case, only the results of these security tests may be inspected. For information, the requirements expected to be tested (or closely related to tests expected to be performed) during the security certification, are marked with a “\*” in this Appendix.

The numbered requirements refer to the Annex corpus, while the other requirements refer to the other appendixes (e.g. PIC\_001 refers to requirement PIC\_001 of Appendix 3 Pictograms).

This Appendix considers separately the type approval of the motion sensor, of the vehicle unit, and of the external GNSS facility as components of the recording equipment. Each component will get its own type approval certificate in which the other compatible components will be indicated. The functional test of the motion sensor (or external GNSS facility) is done together with the vehicle unit and vice versa.

Interoperability between every model of motion sensor (resp. external GNSS facility) and every model of vehicle unit is not required. In that case the type approval for a motion sensor (resp. external GNSS facility) can be granted only in combination with the type approval of the relevant vehicle unit and vice versa.

## References

The following references are used in this Appendix:

- IEC 60068-2-1: Environmental testing - Part 2-1: Tests - Test A: Cold
- IEC 60068-2-2: Basic environmental testing procedures; part 2: tests; tests B: dry heat (sinusoidal).
- IEC 60068-2-6: Environmental testing - Part 2: Tests - Test Fc: Vibration
- IEC 60068-2-14: Environmental testing; Part 2-14 : Tests; Test N: Change of temperature
- IEC 60068-2-27: Environmental testing. Part 2: Tests. Test Ea and guidance: Shock
- IEC 60068-2-30: Environmental testing - Part 2-30: Tests - Test Db: Damp heat, cyclic (12 h + 12 h cycle)
- IEC 60068-2-64: Environmental testing - Part 2-64: Tests - Test Fh: Vibration, broadband random and guidance
- IEC 60068-2-78 Environmental testing - Part 2-78: Tests - Test Cab: Damp heat, steady state

- ISO 16750-3 – Mechanical loads (2012-12)
- ISO 16750-4 - Climatic loads(2010-04).
- ISO 20653: Road vehicles – Degree of protection (IP code) – Protection of electrical equipment against foreign objects, water and access
- ISO 10605 :2008 + Technical Corrigendum :2010 + AMD1 :2014 Road vehicles - Test methods for electrical disturbances from electrostatic discharge
- ISO 7637-1 :2002 + AMD1 : 2008 Road vehicles - Electrical disturbances from conduction and coupling - Part 1: Definitions and general considerations.
- ISO 7637-2 Road vehicles - Electrical disturbances from conduction and coupling - Part 2: Electrical transient conduction along supply lines only.
- ISO 7637-3 Road vehicles - Electrical disturbances from conduction and coupling - Part 3: Electrical transient transmission by capacitive and inductive coupling via lines other than supply lines.
- ISO/IEC 7816-1 Identification cards - Integrated circuit(s) cards with contacts - Part 1: Physical characteristics..
- ISO/IEC 7816-2 Information technology - Identification cards - Integrated circuit(s) cards with contacts - Part 2: Dimensions and location of the contacts.
- ISO/IEC 7816-3 Information technology - Identification cards - Integrated circuit(s) cards with contacts - Part 3: Electronic signals and transmission protocol.
- ISO/IEC 10373-1 :2006 + AMD1 :2012 Identification cards - Test methods - Part 1: General characteristics
- ISO/IEC 10373-3 :2010 + Technical Corrigendum :2013 Identification cards - Test methods - Part 3: Integrated circuit cards with contacts and related interface devices
- ISO 16844-3:2004, Cor 1:2006 Road vehicles – Tachograph systems – Part 3: Motion sensor interface (with vehicle units).
- ISO 16844-4 Road vehicles - Tachograph systems - Part 4: CAN interface
- ISO 16844-6 Road vehicles - Tachograph systems - Part 6: Diagnostics
- ISO 16844-7 Road vehicles - Tachograph systems - Part 7: Parameters
- ISO 534 Paper and board -- Determination of thickness, density and specific volume
- UN ECE R10 Uniform provisions concerning the approval of vehicles with regard to electromagnetic compatibility (United Nation Economic Commission for Europe)

2. Vehicle unit functional tests

<i>No</i>	<i>Test</i>	<i>Description</i>	<i>Related requirements</i>
<b>1</b>	<b>Administrative examination</b>		
1.1	Documentation	Correctness of documentation	
1.2	Manufacturer test results	Results of manufacturer test performed during integration. Paper demonstrations.	88, 89,91
<b>2</b>	<b>Visual inspection</b>		



<i>No</i>	<i>Test</i>	<i>Description</i>	<i>Related requirements</i>
2.1	Compliance with documentation		
2.2	Identification / markings		224 to 226
2.3	Materials		219 to 223
2.4	Sealing		398, 401 to 405
2.5	External interfaces		
<b>3</b>	<b>Functional tests</b>		
3.1	Functions provided		03, 04, 05, 07, 382,
3.2	Modes of operation		09 to 11*, 132, 133
3.3	Functions and data access rights		12* 13*, , 382, 383, 386 to 389
3.4	Monitoring cards insertion and withdrawal		15, 16, 17, 18, 19*, 20*, 132
3.5	Speed and distance measurement		21 to 31
3.6	Time measurement (test performed at 20°C)		38 to 43
3.7	Monitoring driver activities		44 to 53, 132
3.8	Monitoring driving status		54, 55, 132
3.9	Manual entries		56 to 62
3.10	Company locks management		63 to 68
3.11	Monitoring control activities		69, 70
3.12	Detection of events and/or faults		71 to 88 132
3.13	Equipment identification data		93*, 94*, 97, 100
3.14	Driver card insertion and withdrawal data		102* to 104*
3.15	Driver activity data		105* to 107*
3.16	Places and positions data		108* to 112*
3.17	Odometer data		113* to 115*
3.18	Detailed speed data		116*
3.19	Events data		117*
3.20	Faults data		118*
3.21	Calibration data		119* to 121*

<i>No</i>	<i>Test</i>	<i>Description</i>	<i>Related requirements</i>
3.22	Time adjustment data		124*, 125*
3.23	Control activity data		126*, 127*
3.24	Company locks data		128*
3.25	Download activity data		129*
3.26	Specific conditions data		130*, 131*
3.27	Recording and storing on tachographs cards		134, 135., 136*, 137*, 139*, 140, 141  142, 143, 144*, 145*, 146*, 147, 148
3.28	Displaying		90, 132,  149 to 166,  PIC_001, DIS_001
3.29	Printing		90, 132,  167 to 179, PIC_001, PRT_001 to PRT_014
3.30	Warning		132, 180 to 189,  PIC_001
3.31	Data downloading to external media		90, 132, 190 to 194
3.32	Remote communication for targeted roadside checks		195 to 197
3.33	Output data to additional external devices		198, 199
3.34	Calibration		202 to 206*, 383, 384, 386 to 391
3.35	Roadside calibration checking		207 to 209
3.36	Time adjustment		210 to 212*
3.37	Non-interference of additional functions		06, 425
3.38	Motion sensor interface		02, 122
3.39	External GNSS facility		03, 123

<i>No</i>	<i>Test</i>	<i>Description</i>	<i>Related requirements</i>
3.40		Verify that the VU detects, records and stores the event(s) and/or fault(s) defined by the VU manufacturer when a paired motion sensor reacts to magnetic fields disturbing vehicle motion detection.	217
3.41		Cypher suite and standardized domain parameters	CSM_48, CSM_50
<b>4 Environmental tests</b>			
4.1	Temperature	<p>Verify functionality through:</p> <p>Test according to ISO 16750-4, Chapter 5.1.1.2: Low temperature operation test (72 h @ -20 °C)</p> <p>This test refers to IEC 60068-2-1 : Environmental testing - Part 2-1: Tests - Test A: Cold</p> <p>Test according to ISO 16750-4: Chapter 5.1.2.2: High temperature operation test (72 h at 70 °C)</p> <p>This test refers to IEC 60068-2-2: Basic environmental testing procedures; part 2: tests; tests B: dry heat</p> <p>Test according to ISO 16750-4: Chapter 5.3.2: Rapid change of temperature with specified transition duration (-20°C/70 °C, 20 cycles, dwell time 2h at each temperature)</p> <p>A reduced set of tests (among those defined in section 3 of this table) can be carried out at the lower temperature, the higher temperature and during the temperature cycles</p>	213
4.2	Humidity	Verify that the vehicle unit can withstand a cyclic damp (heat test) through IEC IEC 60068-2-30, test Db, six 24 hours cycles, each temperature varying from +25°C to + 55°C and a relative humidity of 97% at + 25°C and equal to 93% at +55°C	214

<i>No</i>	<i>Test</i>	<i>Description</i>	<i>Related requirements</i>
4.3	Mechanical	<p>1. Sinusoidal vibrations.</p> <p>verify that the vehicle unit can withstand sinusoidal vibrations with the following characteristics:</p> <p>constant displacement between 5 and 11 Hz: 10mm peak</p> <p>constant acceleration between 11 and 300 Hz: 5g</p> <p>This requirement is verified through IEC 60068-2-6, test Fc, with a minimum test duration of 3x12 hours (12 hours per axis)</p> <p>ISO 16750-3 does not require a sinusoidal vibration test for devices located in the decoupled vehicle cab.</p> <p>2. Random vibrations:</p> <p>Test according to ISO 16750-3: Chapter 4.1.2.8: Test VIII: Commercial vehicle, decoupled vehicle cab</p> <p>Random vibration test, 10...2000 Hz, RMS vertical 21.3 m/s<sup>2</sup>, RMS longitudinal 11.8 m/s<sup>2</sup>, RMS lateral 13.1 m/s<sup>2</sup>, 3 axes, 32 h per axis, including temperature cycle -20...70°C.</p> <p>This test refers to IEC 60068-2-64: Environmental testing - Part 2-64: Tests - Test Fh: Vibration, broadband random and guidance</p> <p>3. Shocks:</p> <p>mechanical shock with 3g half sinus according ISO 16750.</p> <p>The tests described above are performed on different samples of the equipment type being tested.</p>	219
4.4	Protection against water and foreign bodies	<p>Test according to ISO 20653: Road vehicles – Degree of protection (IP code) – Protection of electrical equipment against foreign objects, water and access (No change in parameters); Minimum value IP 40</p>	220, 221
4.5	Over-voltage protection	<p>Verify that the vehicle unit can withstand a power supply of:</p> <p>24 V versions: 34V at +40°C 1 hour</p> <p>12V versions: 17V at +40°C 1 hour</p> <p>( ISO 16750-2)</p>	216

---

<i>No</i>	<i>Test</i>	<i>Description</i>	<i>Related requirements</i>
4.6	Reverse polarity protection	Verify that the vehicle unit can withstand an inversion of its power supply (ISO 16750-2)	216
4.7	Short-circuit protection	Verify that input output signals are protected against short circuits to power supply and ground (ISO 16750-2)	216
<b>5</b>	<b>EMC tests</b>		
5.1	Radiated emissions and susceptibility	Compliance with Regulation ECE R10	218
5.2	Electrostatic discharge	Compliance with ISO 10605 :2008 + Technical Corrigendum :2010 + AMD1 :2014: +/- 4kV for contact and +/- 8kV for air discharge	218

---

<i>No</i>	<i>Test</i>	<i>Description</i>	<i>Related requirements</i>
5.3	Conducted transient susceptibility on power supply	<p>For 24V versions: compliance with ISO 7637-2 + ECE Regulation No. 10 Rev. 3:</p> <p>pulse 1a: <math>V_s = -450V</math> <math>R_i = 50</math> ohms            pulse 2a: <math>V_s = +37V</math> <math>R_i = 2</math> ohms            pulse 2b: <math>V_s = +20V</math> <math>R_i = 0,05</math> ohms            pulse 3a: <math>V_s = -150V</math> <math>R_i = 50</math> ohms            pulse 3b: <math>V_s = +150V</math> <math>R_i = 50</math> ohms            pulse 4: <math>V_s = -16V</math> <math>V_a = -12V</math> <math>t_6 = 100ms</math>            pulse 5: <math>V_s = +120V</math> <math>R_i = 2,2</math> ohms <math>t_d = 250ms</math></p> <p>For 12V versions: compliance with ISO 7637-1 + ECE Regulation No. 10 Rev. 3:</p> <p>pulse 1: <math>V_s = -75V</math> <math>R_i = 10</math> ohms            pulse 2a: <math>V_s = +37V</math> <math>R_i = 2</math> ohms            pulse 2b: <math>V_s = +10V</math> <math>R_i = 0,05</math> ohms            pulse 3a: <math>V_s = -112V</math> <math>R_i = 50</math> ohms            pulse 3b: <math>V_s = +75V</math> <math>R_i = 50</math> ohms            pulse 4: <math>V_s = -6V</math> <math>V_a = -5V</math> <math>t_6 = 15ms</math>            pulse 5: <math>V_s = +65V</math> <math>R_i = 3ohms</math> <math>t_d = 100ms</math></p> <p>Pulse 5 shall be tested only for vehicle units designed to be installed in vehicles for which no external common protection against load dump is implemented</p> <p>For load dump proposal, refer to ISO 16750-2, 4th edition, chapter 4.6.4.</p>	218

## 3. Motion sensor functional tests

<i>No</i>	<i>Test</i>	<i>Description</i>	<i>Related requirements</i>
<b>1.</b>	<b>Administrative examination</b>		
1.1	Documentation	Correctness of documentation	
<b>2.</b>	<b>Visual inspection</b>		
2.1.	Compliance with documentation		
2.2.	Identification / markings		225, 226,

<i>No</i>	<i>Test</i>	<i>Description</i>	<i>Related requirements</i>
2.3	Materials		219 to 223
2.4.	Sealing		398, 401 to 405
<b>3. Functional tests</b>			
3.1	Sensor identification data		95 to 97*
3.2	Motion sensor – vehicle unit pairing		122*, 204
3.3	Motion detection		
	Motion measurement accuracy		30 to 35
3.4	Vehicle unit interface		02
3.5	Check that the motion sensor is immune to constant magnetic field. Alternatively, verify that the motion sensor reacts to constant magnetic fields disturbing vehicle motion detection so that a connected VU can detect, record and store sensor faults		217
<b>4. Environmental tests</b>			
4.1	Operating temperature	<p>Verify functionality (as defined in test No 3.3) in temperature range [-40°C; +135°C] through:</p> <p>IEC 60068-2-1 test Ad, with a test duration of 96 hours at the lowest temperature <math>T_{\min}</math>,</p> <p>IEC 60068-2-2 test Bd, with a test duration of 96 hours at the highest temperature <math>T_{\max}</math></p> <p>Test according to ISO 16750-4: Chapter 5.1.1.2: Low temperature operation test (24 h @ -40 °C)</p> <p>This test refers to IEC 60068-2-1: Environmental testing - Part 2-1: Tests - Test A: Cold IEC 68-2-2 test Bd, with a test duration of 96 hours at the lowest temperature of -40°C.</p> <p>Test according to ISO 16750-4: Chapter 5.1.2.2: High temperature operation test (96 h @ 135 °C)</p> <p>This test refers to IEC 60068-2-2: Basic environmental testing procedures; part 2: tests; tests B: dry heat</p>	213
4.2	Temperature cycles	<p>Test according to ISO 16750-4: Chapter 5.3.2: Rapid change of temperature with specified transition duration (-40°C/135 °C, 20 cycles, dwell time 30 min at each temperature)</p> <p>IEC 60068-2-14: Environmental testing; Part 2-14 : Tests; Test N: Change of temperature</p>	213

<i>No</i>	<i>Test</i>	<i>Description</i>	<i>Related requirements</i>
4.3	Humidity cycles	Verify functionality (as defined in test No. 3.3) through IEC 60068-2-30, test Db, six 24 hours cycles, each temperature varying from +25°C to +55°C and a relative humidity of 97% at + 25°C and equal to 93% at +55°C	214
4.4	Vibration	ISO 16750-3: Chapter 4.1.2.6: Test VI: Commercial vehicle, engine, gearbox  Mixed mode vibration test including a) Sinusoidal vibration test, 20...520 Hz, 11.4 ... 120 m/s <sup>2</sup> , <= 0.5 oct/min b) Random vibration test, 10...2000 Hz, RMS 177 m/s <sup>2</sup>  94 h per axis, including temperature cycle - 20...70°C)  This test refers to IEC 60068-2-80: Environmental testing - Part 2-80: Tests - Test Fi: Vibration - Mixed mode	219
4.5	Mechanical shock	ISO 16750-3: Chapter 4.2.3: Test VI: Test for devices in or on the gearbox  half-sinusoidal shock, acceleration to be agreed in the range 3000...15000 m/s <sup>2</sup> , pulse duration to be agreed, however < 1 ms, number of shocks: to be agreed  This test refers to IEC 60068-2-27: Environmental testing. Part 2: Tests. Test Ea and guidance: Shock	219
4.6	Protection against water and foreign bodies	Test according to ISO 20653: Road vehicles – Degree of protection (IP code) – Protection of electrical equipment against foreign objects, water and access  (Target value IP 64)	220, 221
4.7	Reverse polarity protection	Verify that the motion sensor can withstand an inversion of its power supply	216
4.8	Short circuit protection	Verify that input output signals are protected against short circuits to power supply and ground	216
<b>5. EMC</b>			
5.1	radiated emissions and susceptibility	Verify compliance with Regulation ECE R10	218



<i>No</i>	<i>Test</i>	<i>Description</i>	<i>Related requirements</i>
5.2	Electrostatic discharge	Compliance with ISO 10605 :2008 + Technical Corrigendum :2010 + AMD1 :2014: +/- 4kV for contact and +/- 8kV for air discharge	218
5.3	Conducted transient susceptibility on data lines)	For 24V versions: compliance with ISO 7637-2 + ECE Regulation No. 10 Rev. 3: pulse 1a: $V_s = -450V$ $R_i = 50$ ohms pulse 2a: $V_s = +37V$ $R_i = 2$ ohms pulse 2b: $V_s = +20V$ $R_i = 0,05$ ohms pulse 3a: $V_s = -150V$ $R_i = 50$ ohms pulse 3b: $V_s = +150V$ $R_i = 50$ ohms pulse 4: $V_s = -16V$ $V_a = -12V$ $t_6 = 100ms$ pulse 5: $V_s = +120V$ $R_i = 2,2$ ohms $t_d = 250ms$ For 12V versions: compliance with ISO 7637-1 + ECE Regulation No. 10 Rev. 3: pulse 1: $V_s = -75V$ $R_i = 10$ ohms pulse 2a: $V_s = +37V$ $R_i = 2$ ohms pulse 2b: $V_s = +10V$ $R_i = 0,05$ ohms pulse 3a: $V_s = -112V$ $R_i = 50$ ohms pulse 3b: $V_s = +75V$ $R_i = 50$ ohms pulse 4: $V_s = -6V$ $V_a = -5V$ $t_6 = 15ms$ pulse 5: $V_s = +65V$ $R_i = 3$ ohms $t_d = 100ms$  Pulse 5 shall be tested only for vehicle units designed to be installed in vehicles for which no external common protection against load dump is implemented  For load dump proposal, refer to ISO 16750-2, 4th edition, chapter 4.6.4	218

## 4. Tachograph cards functional tests

Tests according to this Section 4,

- no. 5 'Protocol tests',
- no. 6 'Card structure' and
- no. 7 'Functional tests'

can be performed by the evaluator or certifier during the Common Criteria (CC) security certification process for the chip module.

Tests number 2.3 and 4.2 are the same. These are the mechanical tests of the combination card body and chip module. If one of these components (card body, chip module) is changed, then these tests are necessary.

No	Test	Description	Related requirements
<b>1.</b>	<b>Administrative examination</b>		
1.1	Documentation	Correctness of documentation	
<b>2</b>	<b>Card Body</b>		
2.1	Printed Design	<p>Make sure that all features for protection and visible data are correctly printed on the card and compliant.</p> <div style="border: 1px solid black; padding: 5px;"> <p>[Designator] Annex 1C, chapter 4.1 'Visible data', 227) The front page shall contain: the words "Driver card" or "Control card" or "Workshop card" or "Company card" printed in capital letters in the official language or languages of the Member State issuing the card, according to the type of the card.</p> </div> <div style="border: 1px solid black; padding: 5px;"> <p>[Member State name] Annex 1C, chapter 4.1 'Visible data', 228) The front page shall contain: the name of the Member State issuing the card (optional).</p> </div> <div style="border: 1px solid black; padding: 5px;"> <p>[Sign] Annex 1C, chapter 4.1 'Visible data', 229) The front page shall contain: the distinguishing sign of the Member State issuing the card, printed in negative in a blue rectangle and encircled by 12 yellow stars.</p> </div>	227 to 229, 232, 234 to 236

		<p>[Enumeration]</p> <p>Annex 1C, chapter 4.1 'Visible data', 232)</p> <p>The reverse page shall contain:  an explanation of the numbered items which appear on the front page of the card.</p> <hr/> <p>[Colour]</p> <p>Annex 1C, chapter 4.1 'Visible data', 234)</p> <p>Tachograph cards shall be printed with the following background predominant colours:</p> <ul style="list-style-type: none"> <li>- driver card: white,</li> <li>- workshop card: red,</li> <li>- control card: blue,</li> <li>- company card: yellow.</li> </ul> <hr/> <p>[Security]</p> <p>Annex 1C, chapter 4.1 'Visible data', 235)</p> <p>Tachograph cards shall bear at least the following features for protection of the card body against counterfeiting and tampering:</p> <ul style="list-style-type: none"> <li>- a security design background with fine guilloche patterns and rainbow printing,</li> <li>- at least one two-coloured microprint line.</li> </ul> <hr/> <p>[Markings]</p> <p>Annex 1C, chapter 4.1 'Visible data', 236)</p> <p>Member States may add colours or markings, such as national symbols and security features.</p> <hr/> <p>[Approval mark]</p> <p>Tachograph cards shall contain an approval mark.</p> <p>The approval mark shall be made up of:</p> <ul style="list-style-type: none"> <li>- a rectangle, within which shall be placed the letter 'e' followed by a distinguishing number or letter for the country which has issued the approval,</li> <li>- an approval number corresponding to the number of the approval certificate for a tachograph card, placed at any point within the immediate proximity of this rectangle.</li> </ul>	
--	--	---	--

<p>2. 2</p>	<p>Mechanical Tests</p>	<p>[Card size] Tachograph cards must conform to standard ISO/IEC 7810, Identification cards - Physical characteristics, [5] Dimension of card, [5.1] Card size, [5.1.1] Card dimensions and tolerances, card type ID-1 Unused card</p> <hr/> <p>[Card edges] Tachograph cards must conform to standard ISO/IEC 7810, Identification cards - Physical characteristics, [5] Dimension of card, [5.1] Card size, [5.1.2] Card edges</p> <hr/> <p>[Card construction] Tachograph cards must conform to standard ISO/IEC 7810, Identification cards - Physical characteristics, [6] Card construction</p> <hr/> <p>[Card materials] Tachograph cards must conform to standard ISO/IEC 7810, Identification cards - Physical characteristics, [7] Card materials</p> <hr/> <p>[Bending stiffness] Tachograph cards must conform to standard ISO/IEC 7810, Identification cards - Physical characteristics, [8] Card characteristics, [8.1] Bending stiffness</p>	<p>240, 243 ISO/IEC 7810</p>
-----------------	-------------------------	--	----------------------------------

		<p>[Toxicity]</p> <p>Tachograph cards must conform to standard ISO/IEC 7810, Identification cards - Physical characteristics,</p> <p>[8] Card characteristics,</p> <p>[8.3] Toxicity</p> <hr/> <p>[Resistance to chemicals]</p> <p>Tachograph cards must conform to standard ISO/IEC 7810, Identification cards - Physical characteristics,</p> <p>[8] Card characteristics,</p> <p>[8.4] Resistance to chemicals</p> <hr/> <p>[Card stability]</p> <p>Tachograph cards must conform to standard ISO/IEC 7810, Identification cards - Physical characteristics,</p> <p>[8] Card characteristics,</p> <p>[8.5] Card dimensional stability and warpage with temperature and humidity</p> <hr/> <p>[Light]</p> <p>Tachograph cards must conform to standard ISO/IEC 7810, Identification cards - Physical characteristics,</p> <p>[8] Card characteristics,</p> <p>[8.6] Light</p> <hr/> <p>[Durability]</p> <p>Annex 1C, chapter 4.4 'Environmental and electrical specifications', 241)</p> <p>Tachograph cards shall be capable of operating correctly for a five-year period if used within the environmental and electrical specifications.</p>	
--	--	---	--

		<p>[Peel strength] Tachograph cards must conform to standard ISO/IEC 7810, Identification cards - Physical characteristics, [8] Card characteristics, [8.8] Peel strength</p> <hr/> <p>[Adhesion or blocking] Tachograph cards must conform to standard ISO/IEC 7810, Identification cards - Physical characteristics, [8] Card characteristics, [8.9] Adhesion or blocking</p> <hr/> <p>[Warpage] Tachograph cards must conform to standard ISO/IEC 7810, Identification cards - Physical characteristics, [8] Card characteristics, [8.11] Overall card warpage</p> <hr/> <p>[Resistance to heat] Tachograph cards must conform to standard ISO/IEC 7810, Identification cards - Physical characteristics, [8] Card characteristics, [8.12] Resistance to heat</p> <hr/> <p>[Surface distortions] Tachograph cards must conform to standard ISO/IEC 7810, Identification cards - Physical characteristics, [8] Card characteristics, [8.13] Surface distortions</p>	
--	--	---	--

		<p>[Contamination]</p> <p>Tachograph cards must conform to standard ISO/IEC 7810, Identification cards - Physical characteristics,</p> <p>[8] Card characteristics,</p> <p>[8.14] Contamination and interaction of card components</p>	
2. 3	Mechanical tests with chip module embedded	<p>[Bending]</p> <p>Tachograph cards must conform to standard ISO/IEC 7810:2003/Amd. 1:2009, Identification cards – Physical characteristics, Amendment 1: Criteria for cards containing integrated circuits</p> <p>[9.2] Dynamic bending stress</p> <p>Total number of bending cycles: 4000.</p> <hr/> <p>[Torsion]</p> <p>Tachograph cards must conform to standard ISO/IEC 7810:2003/Amd. 1:2009, Identification cards – Physical characteristics, Amendment 1: Criteria for cards containing integrated circuits</p> <p>[9.3] Dynamic torsional stress</p> <p>Total number of torsion cycles: 4000.</p>	ISO/IEC 7810
<b>3</b>	<b>Module</b>		
3. 1	Module	<p>Module is the chip encapsulation and the contact plate.</p> <hr/> <p>[Surface profile]</p> <p>Tachograph cards must conform to standard ISO/IEC 7816-1:2011, Identification cards - Integrated circuit cards - Part 1: Cards with contacts - Physical characteristics</p> <p>[4.2] Surface profile of contacts</p> <hr/> <p>[Mechanical strength]</p> <p>Tachograph cards must conform to standard ISO/IEC 7816-1:2011, Identification cards - Integrated circuit cards - Part 1: Cards with contacts - Physical characteristics</p> <p>[4.3] Mechanical strength (of a card and contacts)</p>	ISO/IEC 7816

		<p>[Electrical resistance]</p> <p>Tachograph cards must conform to standard ISO/IEC 7816-1:2011, Identification cards - Integrated circuit cards - Part 1: Cards with contacts - Physical characteristics</p> <p>[4.4] Electrical resistance (of contacts)</p>	
		<p>[Dimension]</p> <p>Tachograph cards must conform to standard ISO/IEC 7816-2:2007, Identification cards - Integrated circuit cards - Part 2: Cards with contacts - Dimension and location of the contacts</p> <p>[3] Dimension of the contacts</p>	
		<p>[Location]</p> <p>Tachograph cards must conform to standard ISO/IEC 7816-2:2007, Identification cards - Integrated circuit cards - Part 2: Cards with contacts - Dimension and location of the contacts</p> <p>[4] Number and location of the contacts</p> <p>In case of modules with six contacts, contact 'C4' and 'C8' are not part of this test requirement.</p>	
<b>4</b>	<b>Chip</b>		
4.1	Chip	<p>[Operating temperature]</p> <p>The Tachograph card chip shall operate in an ambient temperature range between -25 °C and +85 °C.</p> <p>[Temperature and humidity]</p> <p>Annex 1C, chapter 4.4 'Environmental and electrical specifications', 241)</p> <p>Tachograph cards shall be capable of operating correctly in all the climatic conditions normally encountered in Community territory and at least in the temperature range -25°C to +70°C with occasional peaks of up to +85°C, "occasional" meaning not more than 4 hours each time and not over 100 times during the life time of the card.</p> <p>The Tachograph cards are exposed in consecutive steps to the following temperatures and humidities for the given time. After each step the Tachograph cards are tested for electrical functionality.</p> <ol style="list-style-type: none"> <li>1. Temperature of – 20 °C for 2 h.</li> <li>2. Temperature of +/- 0 °C for 2 h.</li> <li>3. Temperature of + 20 °C, 50 % RH, for 2 h.</li> <li>4. Temperature of + 50 °C, 50 % RH, for 2 h.</li> <li>5. Temperature of + 70 °C, 50 % RH, for 2 h.</li> </ol> <p>The temperature is increased intermittently to + 85 °C, 50 % RH, for 60 min.</p> <ol style="list-style-type: none"> <li>6. Temperature of + 70 °C, 85 % RH, for 2 h.</li> </ol> <p>The temperature is increased intermittently to + 85 °C, 85 % RH, for 30 min.</p>	<p>241 to 244</p> <p>ECE R10</p> <p>ISO/IEC 7810</p> <p>ISO/IEC 10373</p>



		<p>[Humidity] Annex 1C, chapter 4.4 'Environmental and electrical specifications', 242) Tachograph cards shall be capable of operating correctly in the humidity range 10% to 90%.</p> <p>[Electromagnetic compatibility - EMC] Annex 1C, chapter 4.4 'Environmental and electrical specifications' 244) During operation, Tachograph cards shall conform to ECE R10 related to electromagnetic compatibility.</p> <p>[Static electricity] Annex 1C, chapter 4.4 'Environmental and electrical specifications', 244) During operation, Tachograph cards shall be protected against electrostatic discharges.  Tachograph cards must conform to standard ISO/IEC 7810:2003/Amd. 1:2009, Identification cards – Physical characteristics, Amendment 1: Criteria for cards containing integrated circuits [9.4] Static electricity [9.4.1] Contact IC cards Test voltage: 4000 V.</p> <p>[X-rays]  Tachograph cards must conform to standard ISO/IEC 7810:2003/Amd. 1:2009, Identification cards – Physical characteristics, Amendment 1: Criteria for cards containing integrated circuits  [9.1] X-rays</p> <p>[Ultraviolet light]  ISO/IEC 10373-1:2006, Identification cards - Test methods - Part 1: General characteristics [5.11] Ultraviolet light</p> <p>[3-wheel]  Tachograph cards must conform to standard ISO/IEC 10373-1:2006/Amd. 1:2012, Identification cards - Test methods - Part 1: General characteristics, Amendment 1 [5.22] ICC - Mechanical strength: 3 wheel test for ICCs with contacts</p> <p>[Wrapping]  Tachograph cards must conform to standard MasterCard CQM V2.03:2013 [11.1.3] R-L3-14-8: Wrapping Test Robustness [13.2.1.32] TM-422: Mechanical Reliability: Wrapping Test</p>	
--	--	--	--

4. 2	Mechanical tests chip module embedded in the card body  -> same as 2.3	<p>[Bending]</p> <p>Tachograph cards must conform to standard ISO/IEC 7810:2003/Amd. 1:2009, Identification cards – Physical characteristics, Amendment 1: Criteria for cards containing integrated circuits [9.2] Dynamic bending stress Total number of bending cycles: 4000.</p> <hr/> <p>[Torsion]</p> <p>Tachograph cards must conform to standard ISO/IEC 7810:2003/Amd. 1:2009, Identification cards – Physical characteristics, Amendment 1: Criteria for cards containing integrated circuits [9.3] Dynamic torsional stress Total number of torsion cycles: 4000.</p>	ISO/IEC 7810
<b>5</b>	<b>Protocol tests</b>		
5. 1	ATR	Check that the ATR is compliant	ISO/IEC 7816-3 TCS_14, TCS_17, TCS_18
5. 2	T=0	Check that T=0 protocol is compliant	ISO/IEC 7816-3 TCS_11, TCS_12, TCS_13, TCS_15
5. 3	PTS	Check that the PTS command is compliant by setting T=1 from T=0.	ISO/IEC 7816-3 TCS_12, TCS_19, TCS_20, TCS_21
5. 4	T=1	Check that T=1 protocol is compliant	ISO/IEC 7816-3 TCS_11, TCS_13, TCS_16
<b>6</b>	<b>Card structure</b>		
6. 1		Test that the file structure of the card is compliant by checking the presence of the mandatory files in the card and their access conditions	TCS_22 to TCS_28  TCS_140 to TCS_179
<b>7</b>	<b>Functional tests</b>		
7. 1	Normal processing	<p>Test at least once each allowed usage of each command (ex: test the UPDATE BINARY command with CLA = '00', CLA = '0C' and with different P1,P2 and Lc parameters)</p> <p>Check that the operations have actually been performed in the card (ex: by reading the file the command has been performed on)</p>	TCS_29 to TCS_139

7.2	Error messages	Test at least once each error message (as specified in Appendix 2) for each command  Test at least once every generic error (except '6400' integrity errors checked during security certification)	
7.3	Cypher suite and standardized domain parameters		CSM_48, CSM_50
<b>8</b>	<b>Personalisation</b>		
8.1	Optical personalisation	<p>Annex 1C, chapter 4.1 'Visible data', 230) The front page shall contain: information specific to the card issued.</p> <p>Annex 1C, chapter 4.1 'Visible data', 231) The front page shall contain: dates using a "dd/mm/yyyy" or "dd.mm.yyyy" format (day, month, year).</p> <p>Annex 1C, chapter 4.1 'Visible data', 235) Tachograph cards shall bear at least the following features for protection of the card body against counterfeiting and tampering: - in the area of the photograph, the security design background and the photograph shall overlap.</p>	230, 231, 235

## 5. External GNSS facility tests

No	Test	Description	Related requirements
<b>1.</b>	<b>Administrative examination</b>		
1.1	Documentation	Correctness of documentation	
<b>2.</b>	<b>Visual inspection for external GNSS facility</b>		
2.1.	Compliance with documentation		
2.2.	Identification / markings		224 to 226
2.3	Materials		219 to 223
<b>3.</b>	<b>Functional tests</b>		
3.1	Sensor identification data		98,99

3. 2	External GNSS module – vehicle unit coupling	123, 205	
3. 3	GNSS position	36, 37	
3. 4	Vehicle unit interface when the GNSS receiver is external to the Vehicle Unit	03	
3. 5	Cypher suite and standardized domain parameters	CSM_48, CSM_50	
<b>4.</b>	<b>Environmental tests</b>		
4. 1	Temperature	<p>Verify functionality through:</p> <p>Test according to ISO 16750-4, Chapter 5.1.1.2: Low temperature operation test (72 h @ -20 °C)</p> <p>This test refers to IEC 60068-2-1: Environmental testing - Part 2-1: Tests - Test A: Cold</p> <p>Test according to ISO 16750-4: Chapter 5.1.2.2: High temperature operation test (72 h @ 70 °C)</p> <p>This test refers to IEC 60068-2-2: Basic environmental testing procedures; part 2: tests; tests B: dry heat</p> <p>Test according to ISO 16750-4: Chapter 5.3.2: Rapid change of temperature with specified transition duration (-20°C/70 °C, 20 cycles, dwell time 1 h at each temperature)</p> <p>A reduced set of tests (among those defined in section 3 of this table) can be carried out at the lower temperature, the higher temperature and during the temperature cycles</p>	213
4. 2	Humidity	Verify that the vehicle unit can withstand a cyclic damp (heat test) through IEC 60068-2-30, test Db, six 24 hours cycles, each temperature varying from +25°C to + 55°C and a relative humidity of 97% at + 25°C and equal to 93% at +55°C	214

4. 3	Mechanical	<p>1. Sinusoidal vibrations.</p> <p>verify that the vehicle unit can withstand sinusoidal vibrations with the following characteristics:</p> <p>constant displacement between 5 and 11 Hz: 10mm peak</p> <p>constant acceleration between 11 and 300 Hz: 5g</p> <p>This requirement is verified through IEC 60068-2-6, test Fc, with a minimum test duration of 3x12 hours (12 hours per axis)</p> <p>ISO 16750-3 does not require a sinusoidal vibration test for devices located in the decoupled vehicle cab.</p> <p>2. Random vibrations:</p> <p>Test according to ISO 16750-3: Chapter 4.1.2.8: Test VIII: Commercial vehicle, decoupled vehicle cab</p> <p>Random vibration test, 10...2000 Hz, RMS vertical 21.3 m/s<sup>2</sup>, RMS longitudinal 11.8 m/s<sup>2</sup>, RMS lateral 13.1 m/s<sup>2</sup>, 3 axes, 32 h per axis, including temperature cycle -20...70°C.</p> <p>This test refers to IEC 60068-2-64: Environmental testing - Part 2-64: Tests - Test Fh: Vibration, broadband random and guidance</p> <p>3. Shocks:</p> <p>mechanical shock with 3g half sinus according ISO 16750.</p> <p>The tests described above are performed on different samples of the equipment type being tested.</p>	219
4. 4	Protection against water and foreign bodies	Test according to ISO 20653: Road vehicles – Degree of protection (IP code) – Protection of electrical equipment against foreign objects, water and access (No change in parameters)	220, 221
4. 5	Over-voltage protection	<p>Verify that the vehicle unit can withstand a power supply of:</p> <p>24 V versions: 34V at +40°C 1 hour</p> <p>12V versions: 17V at +40°C 1 hour</p> <p>(ISO 16750-2, chapter 4.3)</p>	216
4. 6	Reverse polarity protection	<p>Verify that the vehicle unit can withstand an inversion of its power supply</p> <p>(ISO 16750-2, chapter 4.7)</p>	216
4. 7	Short-circuit protection	<p>Verify that input output signals are protected against short circuits to power supply and ground</p> <p>(ISO 16750-2, chapter 4.10)</p>	216

5	EMC tests		
5.1	Radiated emissions and susceptibility	Compliance with Regulation ECE R10	218
5.2	Electrostatic discharge	Compliance with ISO 10605 :2008 + Technical Corrigendum :2010 + AMD1 :2014: +/- 4kV for contact and +/- 8kV for air discharge	218
5.3	Conducted transient susceptibility on power supply	<p>For 24V versions: compliance with ISO 7637-2 + ECE Regulation No. 10 Rev. 3:</p> <p>pulse 1a: <math>V_s = -450V</math> <math>R_i = 50</math> ohms  pulse 2a: <math>V_s = +37V</math> <math>R_i = 2</math> ohms  pulse 2b: <math>V_s = +20V</math> <math>R_i = 0,05</math> ohms  pulse 3a: <math>V_s = -150V</math> <math>R_i = 50</math> ohms  pulse 3b: <math>V_s = +150V</math> <math>R_i = 50</math> ohms  pulse 4: <math>V_s = -16V</math> <math>V_a = -12V</math> <math>t_6 = 100ms</math>  pulse 5: <math>V_s = +120V</math> <math>R_i = 2,2</math> ohms <math>t_d = 250ms</math></p> <p>For 12V versions: compliance with ISO 7637-1 + ECE Regulation No. 10 Rev. 3:</p> <p>pulse 1: <math>V_s = -75V</math> <math>R_i = 10</math> ohms  pulse 2a: <math>V_s = +37V</math> <math>R_i = 2</math> ohms  pulse 2b: <math>V_s = +10V</math> <math>R_i = 0,05</math> ohms  pulse 3a: <math>V_s = -112V</math> <math>R_i = 50</math> ohms  pulse 3b: <math>V_s = +75V</math> <math>R_i = 50</math> ohms  pulse 4: <math>V_s = -6V</math> <math>V_a = -5V</math> <math>t_6 = 15ms</math>  pulse 5: <math>V_s = +65V</math> <math>R_i = 3</math> ohms <math>t_d = 100ms</math></p> <p>Pulse 5 shall be tested only for vehicle units designed to be installed in vehicles for which no external common protection against load dump is implemented</p> <p>For load dump proposal, refer to ISO 16750-2, 4th edition, chapter 4.6.4.</p>	218

## 6. Remote communication facility tests

No	Test	Description	Related requirements
<b>1. Administrative examination</b>			
1.1	Documentation	Correctness of documentation	
<b>2. Visual inspection</b>			
2.1.	Compliance with documentation		
2.2.	Identification / markings		225, 226
2.3	Materials		219 to 223
<b>4. Environmental tests</b>			
4.1	Temperature	<p>Verify functionality through:</p> <p>Test according to ISO 16750-4, Chapter 5.1.1.2: Low temperature operation test (72 h @ -20 °C)</p> <p>This test refers to IEC 60068-2-1: Environmental testing - Part 2-1: Tests - Test A: Cold</p> <p>Test according to ISO 16750-4: Chapter 5.1.2.2: High temperature operation test (72 h @ 70 °C)</p> <p>This test refers to IEC 60068-2-2: Basic environmental testing procedures; part 2: tests; tests B: dry heat</p> <p>Test according to ISO 16750-4: Chapter 5.3.2: Rapid change of temperature with specified transition duration (-20°C/70 °C, 20 cycles, dwell time 1 h (?) at each temperature)</p> <p>A reduced set of tests (among those defined in section 3 of this table) can be carried out at the lower temperature, the higher temperature and during the temperature cycles</p>	213
4.4	Protection against water and foreign bodies	Test according to ISO 20653: Road vehicles – Degree of protection (IP code) – Protection of electrical equipment against foreign objects, water and access (targeted value IP40)	220, 221
<b>5 EMC tests</b>			

5.1	Radiated emissions and susceptibility	Compliance with Regulation ECE R10	218
5.2	Electrostatic discharge	Compliance with ISO 10605 :2008 + Technical Corrigendum :2010 + AMD1 :2014: +/- 4kV for contact and +/- 8kV for air discharge	218
5.3	Conducted transient susceptibility on power supply	<p>For 24V versions: compliance with ISO 7637-2 + ECE Regulation No. 10 Rev. 3:</p> <p>pulse 1a: <math>V_s = -450V</math> <math>R_i = 50</math> ohms  pulse 2a: <math>V_s = +37V</math> <math>R_i = 2</math> ohms  pulse 2b: <math>V_s = +20V</math> <math>R_i = 0,05</math> ohms  pulse 3a: <math>V_s = -150V</math> <math>R_i = 50</math> ohms  pulse 3b: <math>V_s = +150V</math> <math>R_i = 50</math> ohms  pulse 4: <math>V_s = -16V</math> <math>V_a = -12V</math> <math>t_6 = 100ms</math>  pulse 5: <math>V_s = +120V</math> <math>R_i = 2,2</math> ohms <math>t_d = 250ms</math></p> <p>For 12V versions: compliance with ISO 7637-1 + ECE Regulation No. 10 Rev. 3:</p> <p>pulse 1: <math>V_s = -75V</math> <math>R_i = 10</math> ohms  pulse 2a: <math>V_s = +37V</math> <math>R_i = 2</math> ohms  pulse 2b: <math>V_s = +10V</math> <math>R_i = 0,05</math> ohms  pulse 3a: <math>V_s = -112V</math> <math>R_i = 50</math> ohms  pulse 3b: <math>V_s = +75V</math> <math>R_i = 50</math> ohms  pulse 4: <math>V_s = -6V</math> <math>V_a = -5V</math> <math>t_6 = 15ms</math>  pulse 5: <math>V_s = +65V</math> <math>R_i = 3</math> ohms <math>t_d = 100ms</math></p> <p>Pulse 5 shall be tested only for vehicle units designed to be installed in vehicles for which no external common protection against load dump is implemented</p> <p>For load dump proposal, refer to ISO 16750-2, 4th edition, chapter 4.6.4.</p>	218



## 7. Paper functional tests

No	Test	Description	Related requirements
<b>1.</b>	<b>Administrative examination</b>		
1.1	Documentation	Correctness of documentation	
<b>2</b>	<b>General Tests</b>		
2.1	Number of characters per line	Visual inspection of printouts.	172
2.2	Minimum character size	Visual inspection of printout and character inspection.	173
2.3	Supported character sets	The printer shall support characters specified in Appendix 1 Chapter 4 "Character sets".	174
2.4	Printouts definition	Check of tachograph type approval and visual inspection of printouts	174
2.5	Legibility and identification of printouts	<p>Inspection of printouts</p> <p>Demonstrated by test reports and test protocols by manufacturer.</p> <p>All homologation number(s) of tachographs with which the printer paper may be used are imprinted on the paper.</p>	175, 177, 178
2.6	Addition of handwritten notes	<p>Visual inspection: Field for signature of the driver is available.</p> <p>Fields for additional other handwritten entries are available.</p>	180
2.7	Additional details on paper faces.	<p>Paper's face and reverse side may feature additional details and information.</p> <p>These additional details and information may not interfere with the legibility of the printouts.</p> <p>Visual inspection.</p>	177, 178
<b>3</b>	<b>Storage Tests</b>		
3.1	Dry Heat	<p>Preconditioning: 16 hours at +23°C ± 2°C / 55% ± 3% relative humidity</p> <p>Test environment: 72 hours at +70 °C ± 2°C</p> <p>Recovery: 16 hours at +23°C ± 2°C / 55% ± 3% relative humidity</p>	<p>176, 178</p> <p>IEC 60068-2-2-Bb</p>

2. 2	Damp Heat	Preconditioning: 16 hours at $+23^{\circ}\text{C} \pm 2^{\circ}\text{C} / 55\% \pm 3\%$ relative humidity Test environment: 144 hours at $+55^{\circ}\text{C} \pm 2^{\circ}\text{C}$ and $93\% \pm 3\%$ r.h. Recovery: 16 hours at $+23^{\circ}\text{C} \pm 2^{\circ}\text{C} / 55\% \pm 3\%$ relative humidity	176, 178 IEC 60068-2-78-Cab
<b>4</b>	<b>Paper In-Service Tests</b>		
4. 1	Humidity resistance background (unprinted paper)	Preconditioning : 16 hours at $+23^{\circ}\text{C} \pm 2^{\circ}\text{C} / 55\% \pm 3\%$ relative humidity Test environment: 144 hours at $+55^{\circ}\text{C} \pm 2^{\circ}\text{C}$ and $93\% \pm 3\%$ r.h. Recovery : 16 hours at $+23^{\circ}\text{C} \pm 2^{\circ}\text{C} / 55\% \pm 3\%$ relative humidity	176, 178 IEC 60068-2-78-Cab
4. 2	Printability	Preconditioning : 24 hours at $+40^{\circ}\text{C} \pm 2^{\circ}\text{C} / 93\% \pm 3\%$ relative humidity Test environment: printout produced at $+23^{\circ}\text{C} \pm 2^{\circ}\text{C}$ Recovery : 16 hours at $+23^{\circ}\text{C} \pm 2^{\circ}\text{C} / 55\% \pm 3\%$ relative humidity	176, 178
4. 3	Heat resistance	Preconditioning : 16 hours at $+23^{\circ}\text{C} \pm 2^{\circ}\text{C} / 55\% \pm 3\%$ relative humidity Test environment: 2 hours at $+70^{\circ}\text{C} \pm 2^{\circ}\text{C}$ , dry heat Recovery : 16 hours at $+23^{\circ}\text{C} \pm 2^{\circ}\text{C} / 55\% \pm 3\%$ relative humidity	176, 178 IEC 60068-2-2-Bb
4. 4	Low temperature resistance	Preconditioning : 16 hours at $+23^{\circ}\text{C} \pm 2^{\circ}\text{C} / 55\% \pm 3\%$ relative humidity Test environment: 24 hours $-20^{\circ}\text{C} \pm 3^{\circ}\text{C}$ , dry cold Recovery : 16 hours at $+23^{\circ}\text{C} \pm 2^{\circ}\text{C} / 55\% \pm 3\%$ relative humidity	176, 178 ISO 60068-2-1-Ab
4. 5	Light resistance	Preconditioning : 16 hours at $+23^{\circ}\text{C} \pm 2^{\circ}\text{C} / 55\% \pm 3\%$ relative humidity Test environment: 100 hours under 5000 Lux illumination at $+23^{\circ}\text{C} \pm 2^{\circ}\text{C} / 55\% \pm 3\%$ relative humidity Recovery : 16 hours at $+23^{\circ}\text{C} \pm 2^{\circ}\text{C} / 55\% \pm 3\%$ relative humidity	176, 178

Legibility criteria for tests 3.x and 4.x :

Printout legibility is assured if optical densities comply with the following limits:

Printed characters: min. 1,0

Background (unprinted paper): max. 0,2

Optical densities of the resulting printouts shall be measured according to DIN EN ISO 534.

Printouts shall show no dimensional changes and remain clearly legible.

## 8. Interoperability tests

No	Test	Description
9.1 Interoperability tests between vehicle units and tachograph cards		
1	Mutual authentication	Check that the mutual authentication between the vehicle unit and the tachograph card runs normally
2	Write/read tests	<p>Execute a typical activity scenario on the vehicle unit. The scenario shall be adapted to the type of card being tested and involve writings in as many EFs as possible in the card</p> <p>Verify through a vehicle unit downloading that all corresponding recordings have been properly made</p> <p>Verify through a card downloading that all corresponding recordings have been properly made</p> <p>Verify through daily printouts that all corresponding recordings can be properly read</p>
9.2 Interoperability tests between vehicle units and motion sensors		
1	Pairing	Check that the pairing between the vehicle units and the motion sensors runs normally
2	Activity tests	<p>Execute a typical activity scenario on the motion sensor. The scenario shall involve a normal activity and creating as many events or faults as possible.</p> <p>Verify through a vehicle unit downloading that all corresponding recordings have been properly made</p> <p>Verify through a card downloading that all corresponding recordings have been properly made</p> <p>Verify through a daily printout that all corresponding recordings can be properly read</p>
9.3 Interoperability tests between vehicle units and external GNSS facilities (when applicable)		
1	Mutual Authentication	Check that the mutual authentication (coupling) between the vehicle unit and the external GNSS module runs normally.
2	Activity tests	<p>Execute a typical activity scenario on the external GNSS. The scenario shall involve a normal activity and creating as many events or faults as possible.</p> <p>Verify through a vehicle unit downloading that all corresponding recordings have been properly made</p> <p>Verify through a card downloading that all corresponding recordings have been properly made</p> <p>Verify through a daily printout that all corresponding recordings can be properly read</p>

## Appendix 10

# SECURITY REQUIREMENTS

This appendix specifies the IT security requirements for the smart tachograph system components (second-generation tachograph).

SEC\_001 The following components of the smart tachograph system shall be security certified according to the Common Criteria scheme:

- vehicle unit
- tachograph card,
- motion sensor,
- external GNSS facility.

SEC\_002 The minimum IT security requirements to be met by each component needing to be security certified shall be defined in a component Protection Profile, according to the Common Criteria scheme.

SEC\_003 The European Commission shall make sure that four Protection Profiles compliant with this Annex are sponsored, developed, approved by the governmental IT security certification bodies organised within the Joint Interpretation Working Group (JIWG) which is supporting the mutual recognition of certificates under the umbrella of the European SOGIS-MRA (Agreement on Mutual Recognition of Information Technology Security Evaluation Certificates) and registered:

- Protection Profile for vehicle unit,
- Protection Profile for tachograph card,
- Protection Profile for motion sensor,
- Protection Profile for external GNSS facility.

The Protection Profile for vehicle unit shall address the cases when the VU is designed to be used or not with an external GNSS facility. In the former case, the security requirements of the external GNSS facility are provided in the dedicated Protection Profile.

SEC\_004 Component manufacturers shall refine and complete the appropriate component Protection Profile as necessary, without amending or deleting existing threats, objectives, procedural means and security enforcing functions specifications, in order to build a Security Target against which they shall seek the security certification of the component.

SEC\_005 Strict conformance of such specific Security Target with the corresponding Protection Profile must be stated during the evaluation process.

SEC\_006 The assurance level for each Protection Profile shall be EAL4 augmented by the assurance components ATE\_DPT.2 and AVA\_VAN.5.

## Appendix 11 Common security mechanisms

### TABLE OF CONTENTS

<b>Preamble</b>	<b>424</b>
<b>PART A FIRST-GENERATION TACHOGRAPH SYSTEM</b>	<b>425</b>
<b>1. Introduction</b>	<b>425</b>
1.1. .... References	425
1.2. .... Notations and abbreviated terms	425
<b>2. Cryptographic systems and algorithms</b>	<b>427</b>
2.1. .... Cryptographic systems	427
2.2. .... Cryptographic algorithms	427
2.2.1 RSA algorithm .....	427
2.2.2 Hash algorithm.....	427
2.2.3 Data Encryption Algorithm.....	427
<b>3. Keys and certificates</b>	<b>427</b>
3.1. .... Keys generation and distribution	427
3.1.1 RSA Keys generation and distribution.....	427
3.1.2 RSA Test keys .....	429
3.1.3 Motion sensor keys .....	429
3.1.4 T-DES session keys generation and distribution.....	429
3.2. .... Keys	429
3.3. .... Certificates	429
3.3.1 Certificates content .....	429
3.3.2 Certificates issued .....	432
3.3.3 Certificate verification and unwrapping.....	432
<b>4. Mutual authentication mechanism</b>	<b>433</b>
<b>5. VU-Cards data transfer confidentiality, integrity and authentication mechanisms</b>	<b>436</b>
5.1. .... Secure Messaging	436
5.2. .... Treatment of Secure Messaging errors	438
5.3. .... Algorithm to compute Cryptographic Checksums	438
5.4. .... Algorithm to compute cryptograms for confidentiality DOs	439
<b>6. Data download digital signature mechanisms</b>	<b>439</b>
6.1. .... Signature generation	439

6.2. .... Signature verification	439
<b>PART BSECOND-GENERATION TACHOGRAPH SYSTEM</b>	<b>441</b>
<b>7. Introduction</b>	<b>441</b>
7.1. .... References	441
7.2. .... Notations and Abbreviations	442
7.3. .... Definitions	443
<b>8. Cryptographic systems and algorithms</b>	<b>444</b>
8.1. .... Cryptographic Systems	444
8.2. .... Cryptographic Algorithms	445
8.2.1 Symmetric Algorithms.....	445
8.2.2 Asymmetric Algorithms and Standardized Domain Parameters.....	445
8.2.3 Hashing algorithms .....	446
8.2.4 Cipher Suites.....	446
<b>9. Keys and Certificates</b>	<b>446</b>
9.1. .... Asymmetric Key Pairs and Public Key Certificates	446
9.1.1 General.....	446
9.1.2 European Level .....	447
9.1.3 Member State Level.....	448
9.1.4 Equipment Level: Vehicle Units.....	449
9.1.5 Equipment Level: Tachograph Cards.....	450
9.1.6 Equipment Level: External GNSS Facilities.....	452
9.1.7 Overview: Certificate Replacement .....	453
9.2. .... Symmetric Keys	455
9.2.1 Keys for Securing VU – Motion Sensor Communication.....	455
9.2.2 Keys for Securing DSRC Communication .....	460
9.3. .... Certificates	463
9.3.1 General.....	463
9.3.2 Certificate Content .....	464
9.3.3 Requesting Certificates .....	466
<b>10. VU- Card Mutual Authentication and Secure Messaging</b>	<b>467</b>
10.1. .. General	467
10.2. .. Mutual Certificate Chain Verification	467
10.2.1 Card Certificate Chain Verification by VU.....	467

10.2.2	VU Certificate Chain Verification by Card .....	470
10.3	.. VU Authentication	473
10.4	.. Chip Authentication and Session Key Agreement	475
10.5	.. Secure Messaging	476
10.5.1	General.....	476
10.5.2	Secure Message Structure .....	477
10.5.3	Secure Messaging Session Abortion.....	480
<b>11.</b>	<b>VU – External GNSS Facility Coupling, Mutual Authentication and Secure Messaging</b>	<b>483</b>
11.1	.. General	483
11.2	.. VU and External GNSS Facility Coupling	483
11.3	.. Mutual Certificate Chain Verification	484
11.3.1	General.....	484
11.3.2	During VU – EGF Coupling .....	484
11.3.3	During Normal Operation .....	485
11.4	.. VU Authentication, Chip Authentication and Session Key Agreement	486
11.5	.. Secure Messaging	486
<b>12.</b>	<b>VU – Motion Sensor Pairing and Communication</b>	<b>487</b>
12.1	.. General	487
12.2	.. VU – Motion Sensor Pairing Using Different Key Generations	487
12.3	.. VU – Motion Sensor Pairing and Communication using AES	488
12.4	.. VU – Motion Sensor Pairing For Different Equipment Generations	490
<b>13.</b>	<b>Security for Remote Communication over DSRC</b>	<b>491</b>
13.1	.. General	491
13.2	.. Tachograph Payload Encryption and MAC Generation	492
13.3	.. Verification and Decryption of Tachograph Payload	492
<b>14.</b>	<b>Signing Data Downloads and Verifying Signatures</b>	<b>494</b>
14.1	.. General	494
14.2	.. Signature generation	494
14.3	.. Signature verification	494

## Preamble

This Appendix specifies the security mechanisms ensuring

- mutual authentication between different components of the tachograph system.
- confidentiality, integrity, authenticity and/or non-repudiation of data transferred between different components of the tachograph system or downloaded to external storage media.

This Appendix consists of two parts. Part A defines the security mechanisms for the first-generation tachograph system (digital tachograph). Part B defines the security mechanisms for the second-generation tachograph system (smart tachograph).

The mechanisms specified in Part A of this Appendix shall apply if at least one of the components of the tachograph system involved in a mutual authentication and/or data transfer process is of the first generation.

The mechanisms specified in Part B of this Appendix shall apply if both components of the tachograph system involved in the mutual authentication and/or data transfer process are of the second generation.

Appendix 15 provides more information regarding the use of first generation components in combination with second-generation components.



## PART A FIRST-GENERATION TACHOGRAPH SYSTEM

### 1. Introduction

#### 1.1. References

The following references are used in this Appendix:

SHA-1	National Institute of Standards and Technology (NIST). <i>FIPS Publication 180-1 : Secure Hash Standard</i> . April 1995.
PKCS1	RSA Laboratories. PKCS # 1 : <i>RSA Encryption Standard</i> . Version 2.0. October 1998.
TDES	National Institute of Standards and Technology (NIST). <i>FIPS Publication 46-3 : Data Encryption Standard</i> . Draft 1999.
TDES-OP	ANSI X9.52, Triple Data Encryption Algorithm Modes of Operation. 1998.
ISO/IEC 7816-4	Information Technology – Identification cards – Integrated circuit(s) cards with contacts – Part 4: Interindustry commands for interexchange. First edition: 1995 + Amendment 1: 1997.
ISO/IEC 7816-6	Information Technology – Identification cards – Integrated circuit(s) cards with contacts – Part 6: Interindustry data elements. First edition: 1996 + Cor 1: 1998.
ISO/IEC 7816-8	Information Technology – Identification cards – Integrated circuit(s) cards with contacts – Part 8: Security related interindustry commands. First edition 1999.
ISO/IEC 9796-2	Information Technology – Security techniques – Digital signature schemes giving message recovery – Part 2: Mechanisms using a hash function. First edition: 1997.
ISO/IEC 9798-3	Information Technology – Security techniques – Entity authentication mechanisms – Part 3: Entity authentication using a public key algorithm. Second edition 1998.
ISO 16844-3	Road vehicles – Tachograph systems – Part 3: Motion sensor interface.

#### 1.2. Notations and abbreviated terms

The following notations and abbreviated terms are used in this Appendix:

$(K_a, K_b, K_c)$	a key bundle for use by the Triple Data Encryption Algorithm,
CA	Certification Authority,
CAR	Certification Authority Reference,
CC	Cryptographic Checksum,
CG	Cryptogram,
CH	Command Header,
CHA	Certificate Holder Authorisation,
CHR	Certificate Holder Reference,
D()	Decryption with DES,
DE	Data Element,
DO	Data Object,

<i>d</i>	RSA private key, private exponent,
<i>e</i>	RSA public key, public exponent,
E()	Encryption with DES,
EQT	Equipment,
<i>Hash()</i>	hash value, an output of <i>Hash</i> ,
<i>Hash</i>	hash function,
KID	Key Identifier,
Km	TDES key. Master Key defined in ISO 16844-3.
Km <sub>VU</sub>	TDES key inserted in vehicle units.
Km <sub>WC</sub>	TDES key inserted in workshop cards.
<i>m</i>	message representative, an integer between 0 and <i>n</i> -1,
<i>n</i>	RSA keys, modulus,
PB	Padding Bytes,
PI	Padding Indicator byte (for use in Cryptogram for confidentiality DO),
PV	Plain Value,
<i>s</i>	signature representative, an integer between 0 and <i>n</i> -1,
SSC	Send Sequence Counter,
SM	Secure Messaging,
TCBC	TDEA Cipher Block Chaining Mode of Operation
TDEA	Triple Data Encryption Algorithm,
TLV	Tag Length Value,
VU	Vehicle Unit,
X.C	the certificate of user X issued by a certification authority,
X.CA	a certification authority of user X,
X.CA.PK <sub>o</sub> X.C	the operation of unwrapping a certificate to extract a public key. It is an infix operator, whose left operand is the public key of a certification authority, and whose right operand is the certificate issued by that certification authority. The outcome is the public key of the user X whose certificate is the right operand,
X.PK	RSA public key of a user X,
X.PK[I]	RSA encipherment of some information I, using the public key of user X,
X.SK	RSA private key of a user X,
X.SK[I]	RSA encipherment of some information I, using the private key of user X,
'xx'	an Hexadecimal value,
	concatenation operator.

## 2. Cryptographic systems and algorithms

### 2.1. Cryptographic systems

CSM\_001 Vehicle units and tachograph cards shall use a classical RSA public-key cryptographic system to provide the following security mechanisms:

- authentication between vehicle units and cards,
- transport of Triple-DES session keys between vehicle units and tachograph cards,
- digital signature of data downloaded from vehicle units or tachograph cards to external media.

CSM\_002 Vehicle units and tachograph cards shall use a Triple DES symmetric cryptographic system to provide a mechanism for data integrity during user data exchange between vehicle units and tachograph cards, and to provide, where applicable, confidentiality of data exchange between vehicle units and tachograph cards.

### 2.2. Cryptographic algorithms

#### 2.2.1 RSA algorithm

CSM\_003 The RSA algorithm is fully defined by the following relations:

$$X.SK[m] = s = m^d \bmod n$$

$$X.PK[s] = m = s^e \bmod n$$

A more comprehensive description of the RSA function can be found in reference [PKCS1]. Public exponent,  $e$ , for RSA calculations is an integer between 3 and  $n-1$  satisfying  $\gcd(e, \text{lcm}(p-1, q-1))=1$ .

#### 2.2.2 Hash algorithm

CSM\_004 The digital signature mechanisms shall use the SHA-1 hash algorithm as defined in reference [SHA-1].

#### 2.2.3 Data Encryption Algorithm

CSM\_005 DES based algorithms shall be used in Cipher Block Chaining mode of operation.

## 3. Keys and certificates

### 3.1. Keys generation and distribution

#### 3.1.1 RSA Keys generation and distribution

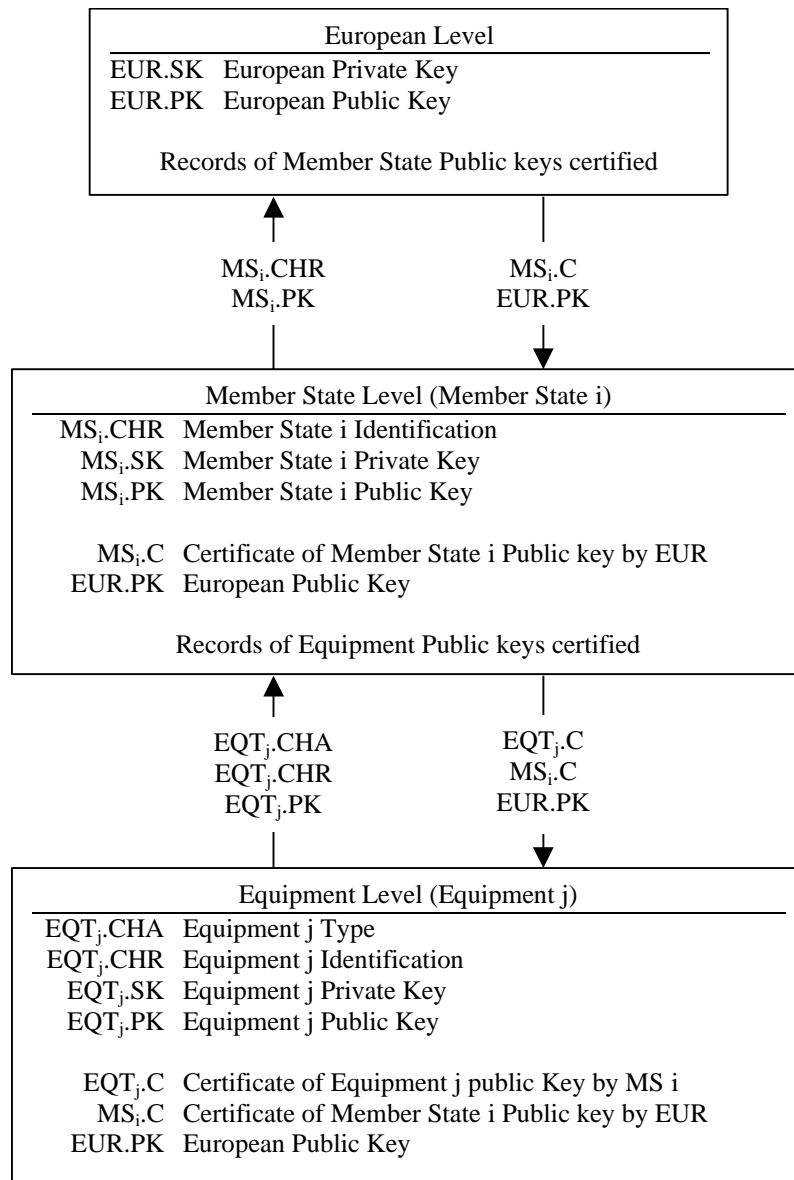
CSM\_006 RSA keys shall be generated through three functional hierarchical levels:

- European level,
- Member State level,
- Equipment level.

CSM\_007 At European level, a single European key pair (EUR.SK and EUR.PK) shall be generated. The European private key shall be used to certify the Member States public keys. Records of all certified keys shall be kept. These tasks shall be handled by a European Certification Authority, under the authority and responsibility of the European Commission.

- CSM\_008 At Member State level, a Member State key pair (MS.SK and MS.PK) shall be generated. Member States public keys shall be certified by the European Certification Authority. The Member State private key shall be used to certify public keys to be inserted in equipment (vehicle unit or tachograph card). Records of all certified public keys shall be kept with the identification of the equipment to which it is intended. These tasks shall be handled by a Member State Certification Authority. A Member State may regularly change its key pair.
- CSM\_009 At equipment level, one single key pair (EQT.SK and EQT.PK) shall be generated and inserted in each equipment. Equipment public keys shall be certified by a Member State Certification Authority. These tasks may be handled by equipment manufacturers, equipment personalisers or Member State authorities. This key pair is used for authentication, digital signature and encipherment services
- CSM\_010 Private keys confidentiality shall be maintained during generation, transport (if any) and storage.

The following picture summarises the data flow of this process:



### 3.1.2 RSA Test keys

CSM\_011 For the purpose of equipment testing (including interoperability tests) the European Certification Authority shall generate a different single European test key pair and at least two Member State test key pairs, the public keys of which shall be certified with the European private test key. Manufacturers shall insert, in equipment undergoing type approval tests, test keys certified by one of these Member State test keys.

### 3.1.3 Motion sensor keys

The confidentiality of the three Triple DES keys described below shall be appropriately maintained during generation, transport (if any) and storage.

In order to support tachograph components compliant with ISO 16844, the European Certification Authority and the Member State Certification Authorities shall, in addition, ensure the following:

CSM\_036 The European Certification authority shall generate  $K_{mVU}$  and  $K_{mWC}$ , two independent and unique Triple DES keys, and generate  $K_m$  as :  $K_m = K_{mVU} \text{ XOR } K_{mWC}$ . The European Certification Authority shall forward these keys, under appropriately secured procedures, to Member States Certification Authorities at their request.

CSM\_037 Member States Certification Authorities shall:

- use  $K_m$  to encrypt motion sensor data requested by motion sensor manufacturers (data to be encrypted with  $K_m$  is defined in ISO 16844-3),
- forward  $K_{mVU}$  to vehicle unit manufacturers, under appropriately secured procedures, for insertion in vehicle units,
- ensure that  $K_{mWC}$  will be inserted in all workshop cards (SensorInstallationSecData in Sensor\_Installation\_Data elementary file) during card personalisation.

### 3.1.4 T-DES session keys generation and distribution

CSM\_012 Vehicle units and tachograph cards shall, as a part of the mutual authentication process, generate and exchange necessary data to elaborate a common Triple DES session key. This exchange of data shall be protected for confidentiality through an RSA crypt-mechanism.

CSM\_013 This key shall be used for all subsequent cryptographic operations using secure messaging. Its validity shall expire at the end of the session (withdrawal of the card or reset of the card) and/or after 240 use (one use of the key = one command using secure messaging sent to the card and associated response).

## 3.2. Keys

CSM\_014 RSA keys shall have (whatever the level) the following lengths: modulus  $n$  1024 bits, public exponent  $e$  64 bits maximum, private exponent  $d$  1024 bits.

CSM\_015 Triple DES keys shall have the form  $(K_a, K_b, K_a)$  where  $K_a$  and  $K_b$  are independent 64 bits long keys. No parity error detecting bits shall be set.

## 3.3. Certificates

CSM\_016 RSA Public key certificates shall be “non self-descriptive” “Card Verifiable” certificates (Ref.: ISO/IEC 7816-8)

### 3.3.1 Certificates content

CSM\_017 RSA Public key certificates are built with the following data in the following order :

<i>Data</i>	<i>Format</i>	<i>Bytes</i>	<i>Obs</i>
CPI	INTEGER	1	Certificate Profile Identifier ('01' for this version)
CAR	OCTET STRING	8	Certification Authority Reference
CHA	OCTET STRING	<b>7</b>	Certificate Holder Authorisation
EOV	TimeReal	4	Certificate end of validity. Optional, "FF" padded if not used.
CHR	OCTET STRING	8	Certificate Holder Reference
<i>n</i>	OCTET STRING	128	Public key (modulus)
<i>e</i>	OCTET STRING	8	Public Key (public exponent)
<b>164</b>			

**Notes:**

1. The "Certificate Profile Identifier" (CPI) delineates the exact structure of an authentication certificate. It can be used as an equipment internal identifier of a relevant headerlist which describes the concatenation of Data Elements within the certificate.

The headerlist associated with this certificate content is as follows:

'4 D ,	'1 6'	'5F 29'	'0 1'	'4 2'	'0 8'	'5F 4B'	'0 7'	'5F 24'	'0 4'	'5F 20'	'0 8'	'7F 49'	'0 5'	'8 1'	'81 80'	'8 2'	'0 8'
Extended Headerlist Tag	Length of header list	CPI Tag	CPI Length	CAR Tag	CAR Length	CHA Tag	CHA Length	EOV Tag	EOV Length	CHR Tag	CHR Length	Public Key Tag (Constructed)	Length of subsequent DOs	modulus Tag	modulus length	public exponent Tag	public exponent length

2. The "Certification Authority Reference" (CAR) has the purpose of identifying the certificate issuing CA, in such a way that the Data Element can be used at the same time as an Authority Key Identifier to reference the Public Key of the Certification Authority (for coding, see Key Identifier below).

3. The “Certificate Holder Authorisation” (CHA) is used to identify the rights of the certificate holder. It consists of the Tachograph Application ID and of the type of equipment to which the certificate is intended (according to *EquipmentType* data element, ‘00’ for a Member State).
4. The “Certificate Holder Reference” (CHR) has the purpose of identifying uniquely the certificate holder, in such a way that the Data Element can be used at the same time as a Subject Key Identifier to reference the Public Key of the certificate holder.
5. Key Identifiers uniquely identify certificate holder or certification authorities. They are coded as follows:

#### 5.1 Equipment (VU or Card):

<i>Data</i>	<i>Equipment serial number</i>	<i>Date</i>	<i>Type</i>	<i>Manufacturer</i>
<b>Length</b>	4 Bytes	2 Bytes	1 Byte	1 Byte
<b>Value</b>	Integer	mm yy BCD coding	Manufacturer specific	Manufacturer code

In the case of a VU, the manufacturer, when requesting certificates, may or may not know the identification of the equipment in which the keys will be inserted.

In the first case, the manufacturer will send the equipment identification with the public key to its Member State authority for certification. The certificate will then contain the equipment identification, and the manufacturer must ensure that keys and certificate are inserted in the intended equipment. The Key identifier has the form shown above.

In the later case, the manufacturer must uniquely identify each certificate request and send this identification with the public key to its Member State authority for certification. The certificate will contain the request identification. The manufacturer must feed back its Member State authority with the assignment of key to equipment (i.e. certificate request identification, equipment identification) after key installation in the equipment. The key identifier has the following form:

<i>Data</i>	<i>Certificate request serial number</i>	<i>Date</i>	<i>Type</i>	<i>Manufacturer</i>
<b>Length</b>	4 Bytes	2 Bytes	1 Byte	1 Byte
<b>Value</b>	Integer	mm yy BCD coding	‘FF’	Manufacturer code

#### 5.2 Certification Authority:

<i>Data</i>	<i>Authority Identification</i>	<i>Key serial number</i>	<i>Additional info</i>	<i>Identifier</i>
<b>Length</b>	4 Bytes	1 Byte	2 Bytes	1 Byte
<b>Value</b>	1 Byte nation numerical code 3 Bytes nation alphanumerical code	Integer	additional coding (CA specific) ‘FF FF’ if not used	‘01’

The key serial number is used to distinguish the different keys of a Member State, in the case the key is changed.

6. Certificate verifiers shall implicitly know that the public key certified is an RSA key relevant to authentication, digital signature verification and encipherment for confidentiality services (the certificate contains no Object Identifier to specify it).

3.3.2 Certificates issued

CSM\_018 The certificate issued is a digital signature with partial recovery of the certificate content in accordance with ISO/IEC 9796-2 (except for its annex A4), with the “Certification Authority Reference” appended.

$$X.C = X.CA.SK[‘6A’ \parallel C_r \parallel Hash(Cc) \parallel ‘BC’] \parallel C_n \parallel X.CAR$$

With certificate content =  $C_c = C_r \parallel C_n$   
 106 bytes      58 bytes

Notes:

1. This certificate is 194 bytes long.
2. CAR, being hidden by the signature, is also appended to the signature, such that the Public Key of the Certification Authority may be selected for the verification of the certificate.
3. The certificate verifier shall implicitly know the algorithm used by the Certification Authority to sign the certificate.
4. The headerlist associated with this issued certificate is as follows:

‘7F 21’	‘09’	‘5F 37’	‘81 80’	‘5F 38’	‘3 A’	‘4 2’	‘0 8’
CV Certificate Tag (Constructed)	Length of subsequent DOs	Signature Tag	Signature Length	Remainder Tag	Remainder Length	CAR Tag	CAR Length

3.3.3 Certificate verification and unwrapping

Certificate verification and unwrapping consists in verifying the signature in accordance with ISO/IEC 9796-2, retrieving the certificate content and the public key contained :  $X.PK = X.CA.PK \circ X.C$ , and verifying the validity of the certificate.

CSM\_019 It involves the following steps :

Verify signature and retrieve content:

- from X.C retrieve Sign,  $C_n'$  and CAR':  $X.C = \text{Sign} \parallel C_n' \parallel \text{CAR}'$   
 128 Bytes    58 Bytes    8 Bytes
- from CAR' select appropriate Certification Authority Public Key (if not done before through other means)
- open Sign with CA Public Key :  $Sr' = X.CA.PK [\text{Sign}]$ ,
- check  $Sr'$  starts with ‘6A’ and ends with ‘BC’
- compute  $C_r'$  and H' from:  $Sr' = ‘6A’ \parallel C_r' \parallel H' \parallel ‘BC’$



106 Bytes 20 Bytes

- Recover certificate content  $C' = C_r' \parallel C_n'$ ,
- check  $Hash(C') = H'$

If the checks are OK the certificate is a genuine one, its content is  $C'$ .

Verify validity. From  $C'$ :

- if applicable, check End of validity date,

Retrieve and store public key, Key Identifier, Certificate Holder Authorisation and Certificate End of Validity from  $C'$ :

- X.PK =  $n \parallel e$
- X.KID = CHR
- X.CHA = CHA
- X.EOV = EOVS

#### 4. Mutual authentication mechanism

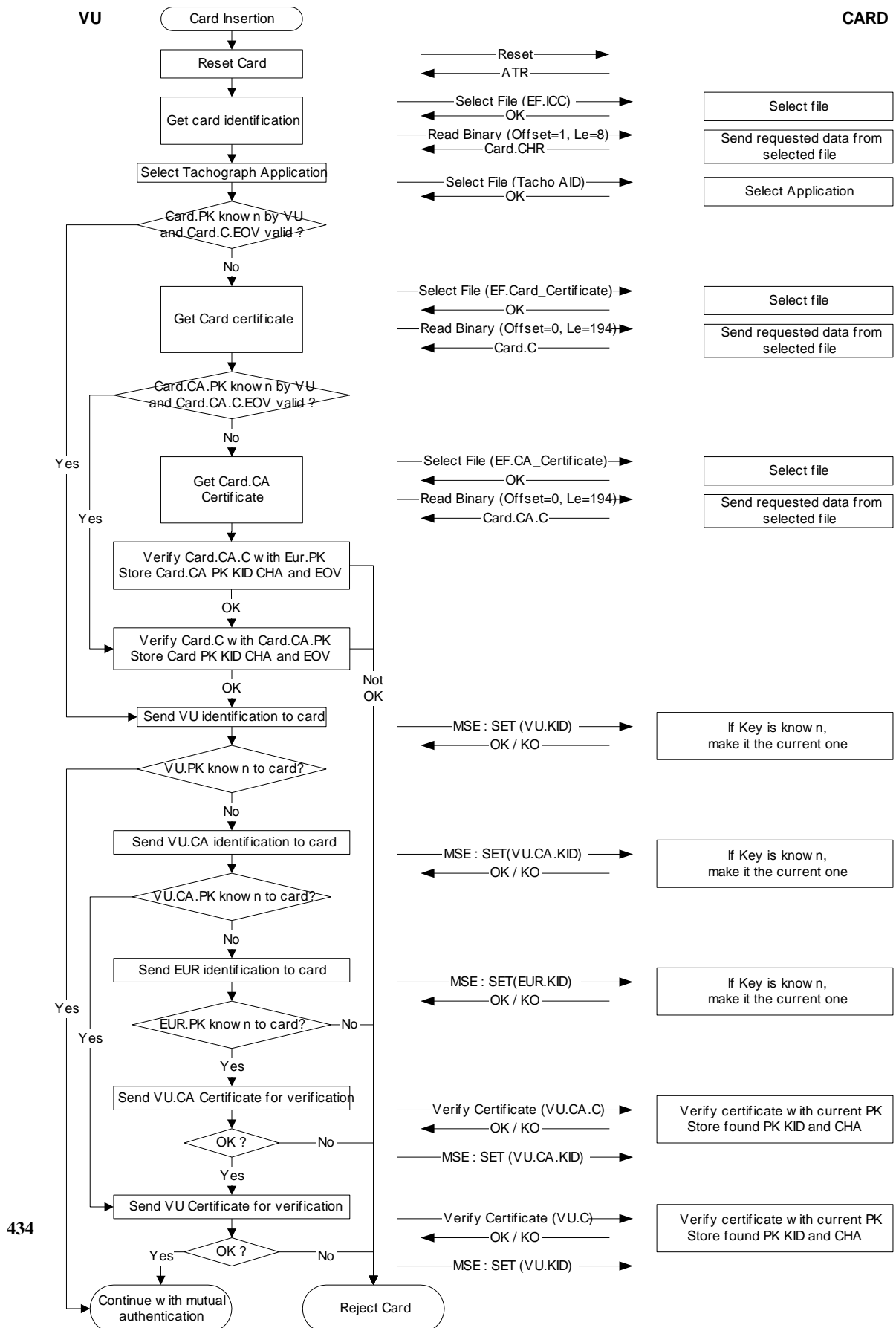
Mutual authentication between cards and VUs is based on the following principle :

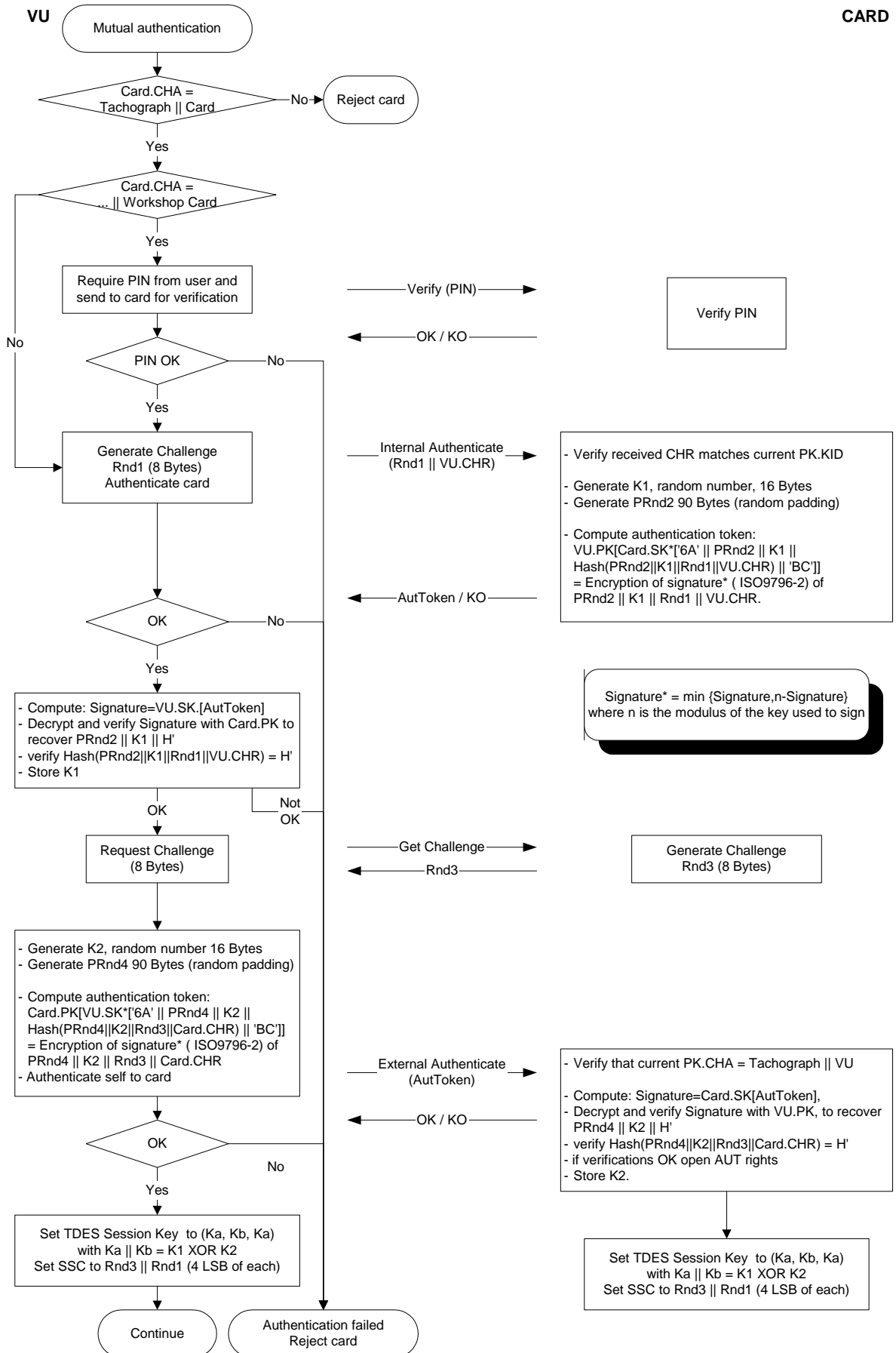
Each party shall demonstrate to the other that it owns a valid key pair, the public key of which has been certified by a Member State certification authority, itself being certified by the European certification authority.

Demonstration is made by signing with the private key a random number sent by the other party, who must recover the random number sent when verifying this signature.

The mechanism is triggered at card insertion by the VU. It starts with the exchange of certificates and unwrapping of public keys, and ends with the setting of a session key.

CSM\_020 The following protocol shall be used (arrows indicate commands and data exchanged (see Appendix 2)):





5. VU-Cards data transfer confidentiality, integrity and authentication mechanisms

5.1. Secure Messaging

- CSM\_021 VU-Cards data transfers integrity shall be protected through Secure Messaging in accordance with references [ISO/IEC 7816-4] and [ISO/IEC 7816-8].
- CSM\_022 When data need to be protected during transfer, a Cryptographic Checksum Data Object shall be appended to the Data Objects sent within the command or the response. The Cryptographic Checksum shall be verified by the receiver.
- CSM\_023 The cryptographic checksum of data sent within a command shall integrate the command header, and all data objects sent ( $\Rightarrow$ CLA = '0C', and all data objects shall be encapsulated with tags in which b1=1).
- CSM\_024 The response status-information bytes shall be protected by a cryptographic checksum when the response contains no data field.
- CSM\_025 Cryptographic checksums shall be 4 Bytes long.

The structure of commands and responses when using secure messaging is therefore the following:

The DOs used are a partial set of the Secure Messaging DOs described in ISO/IEC 7816-4:

Tag	Mnemonic	Meaning
'81'	T <sub>PV</sub>	Plain Value not BER-TLV coded data (to be protected by CC)
'97'	<b>T</b> <sub>LE</sub>	Value of Le in the unsecured command (to be protected by CC)
'99'	T <sub>SW</sub>	Status-Info (to be protected by CC)
'8E'	T <sub>CC</sub>	Cryptographic Checksum
'87'	T <sub>PICG</sub>	Padding Indicator Byte    Cryptogram (Plain Value not coded in BER-TLV)

Given an unsecured command response pair:

Command header				Command body		
CLA	INS	P1	P2	[L <sub>c</sub> field]	[Data field]	[L <sub>e</sub> field]
four bytes				L bytes, denoted as B <sub>1</sub> to B <sub>L</sub>		

Response body	Response trailer	
[Data field]	SW1	SW2
L <sub>r</sub> data bytes	two bytes	

The corresponding secured command response pair is:

Secured command:

<i>Command header (CH)</i>				<i>Command body</i>										
CLA	INS	P1	P2	[New L <sub>c</sub> field]	[New Data field]								[New L <sub>e</sub> field]	
				Length of	T <sub>P</sub>	L <sub>P</sub>	P	T <sub>L</sub>	L <sub>L</sub>	L <sub>e</sub>	T <sub>C</sub>	L <sub>C</sub>	C	'00'
				New Data	v	v	V	E	E		C	C	C	
				field	'8	L <sub>c</sub>	Da	'9	'0	L <sub>e</sub>	'8	'0	C	
					1'		ta	7'	1'		E'	4'	C	
							fi							
							ld							

Data to be integrated in checksum = CH || PB || T<sub>PV</sub> || L<sub>PV</sub> || PV || T<sub>LE</sub> || L<sub>LE</sub> || L<sub>c</sub> || PB

PB = Padding Bytes (80 .. 00) in accordance with ISO-IEC 7816-4 and ISO 9797 method 2.

DOs PV and LE are present only when there is some corresponding data in the unsecured command.

Secured response:

1. Case where response data field is not empty and needs not to be protected for confidentiality:

<i>Response body</i>						<i>Response trailer</i>	
[New Data field]						new SW1 SW2	
T <sub>PV</sub>	L <sub>PV</sub>	PV	T <sub>CC</sub>	L <sub>CC</sub>	CC		
'81'	L <sub>r</sub>	Data field	'8E'	'04'	CC		

Data to be integrated in checksum = T<sub>PV</sub> || L<sub>PV</sub> || PV || PB

2. Case where response data field is not empty and needs to be protected for confidentiality:

<i>Response body</i>						<i>Response trailer</i>	
[New Data field]						new SW1 SW2	
T <sub>PI CG</sub>	L <sub>PI CG</sub>	PI CG	T <sub>CC</sub>	L <sub>CC</sub>	CC		
'87'		PI    CG	'8E'	'04'	CC		

Data to be carried by CG : non BER-TLV coded data and padding bytes.

Data to be integrated in checksum = T<sub>PI CG</sub> || L<sub>PI CG</sub> || PI CG || PB

3. Case where response data field is empty:

<i>Response body</i>						<i>Response trailer</i>	
[New Data field]						new SW1 SW2	
T <sub>SW</sub>	L <sub>SW</sub>	SW	T <sub>CC</sub>	L <sub>CC</sub>	CC		
'99'	'02'	New SW1 SW2	'8E'	'04'	CC		

Data to be integrated in checksum = T<sub>SW</sub> || L<sub>SW</sub> || SW || PB

5.2. Treatment of Secure Messaging errors

CSM\_026 When the tachograph card recognises an SM error while interpreting a command, then the status bytes must be returned without SM. In accordance with ISO/IEC 7816-4, the following status bytes are defined to indicate SM errors:

- '66 88': Verification of Cryptographic Checksum failed,
- '69 87': Expected SM Data Objects missing,
- '69 88': SM Data Objects incorrect.

CSM\_027 When the tachograph card returns status bytes without SM DOs or with an erroneous SM DO, the session must be aborted by the VU.

5.3. Algorithm to compute Cryptographic Checksums

CSM\_028 Cryptographic checksums are built using a retail MACs in accordance with ANSI X9.19 with DES:

- Initial stage: The initial check block  $y_0$  is  $E(K_a, SSC)$ .
- Sequential stage: The check blocks  $y_1, \dots, y_n$  are calculated using  $K_a$ .
- Final stage: The cryptographic checksum is calculated from the last check block  $y_n$  as follows:  $E(K_a, D(K_b, y_n))$ .

where  $E()$  means encryption with DES, and  $D()$  means decryption with DES.

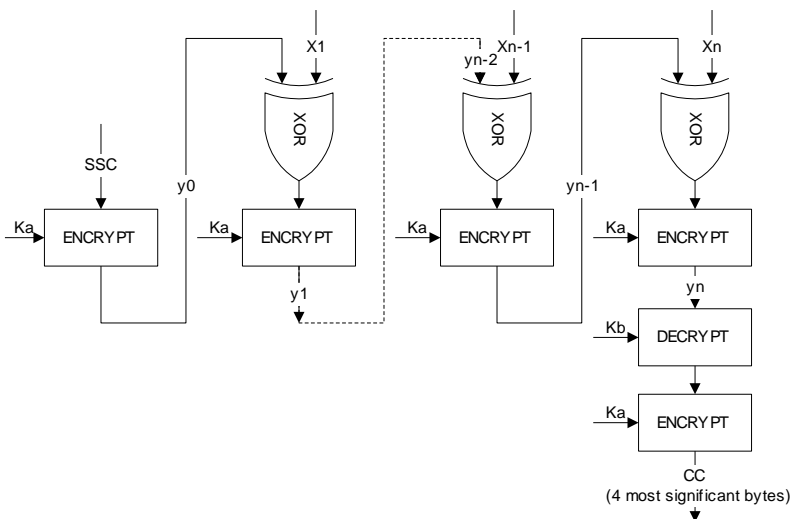
The four most significant bytes of the cryptographic checksum are transferred

CSM\_029 The Send Sequence Counter (SSC) shall be initiated during key agreement procedure to:

Initial SSC :  $Rnd3$  (4 least significant bytes) ||  $Rnd1$  (4 least significant bytes).

CSM\_030 The Send Sequence Counter shall be increased by 1 each time before a MAC is calculated (i.e. the SSC for the first command is Initial SSC + 1, the SSC for the first response is Initial SSC + 2).

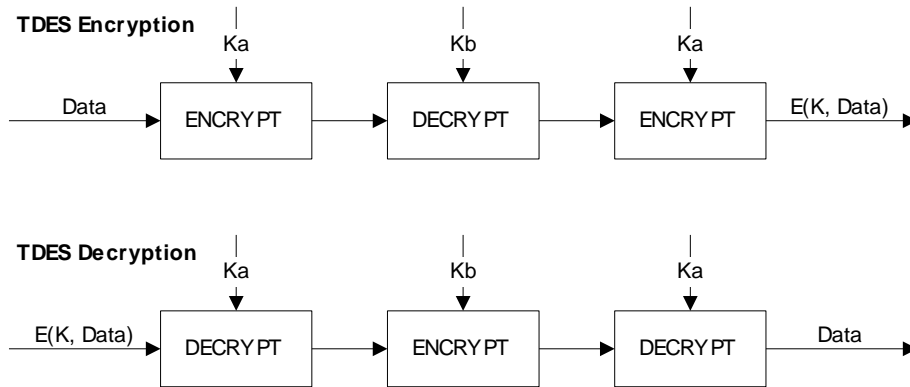
The following figure shows the calculation of the retail MAC:



5.4. Algorithm to compute cryptograms for confidentiality DOs

CSM\_031 Cryptograms are computed using TDEA in TCBC mode of operation in accordance with references [TDES] and [TDES-OP] and with the Null vector as Initial Value block.

The following figure shows the application of keys in TDES:



6. Data download digital signature mechanisms

CSM\_032 The Intelligent Dedicated Equipment (IDE) stores data received from an equipment (VU or card) during one download session within one physical data file. This file must contain the certificates MS<sub>i</sub>.C and EQT.C. The file contains digital signatures of data blocks as specified in Appendix 7 Data Downloading Protocols.

CSM\_033 Digital signatures of downloaded data shall use a digital signature scheme with appendix such, that downloaded data may be read without any decipherment if desired.

6.1. Signature generation

CSM\_034 Data signature generation by the equipment shall follow the signature scheme with appendix defined in reference [PKCS1] with the SHA-1 hash function :

$$\text{Signature} = \text{EQT.SK}[\text{'00' || '01' || PS || '00' || DER(SHA-1(Data))}]$$

PS = Padding string of octets with value 'FF' such that length is 128.

DER(SHA-1(M)) is the encoding of the algorithm ID for the hash function and the hash value into an ASN.1 value of type `DigestInfo` (distinguished encoding rules):

'30' || '21' || '30' || '09' || '06' || '05' || '2B' || '0E' || '03' || '02' || '1A' || '05' || '00' || '04' || '14' || Hash Value.

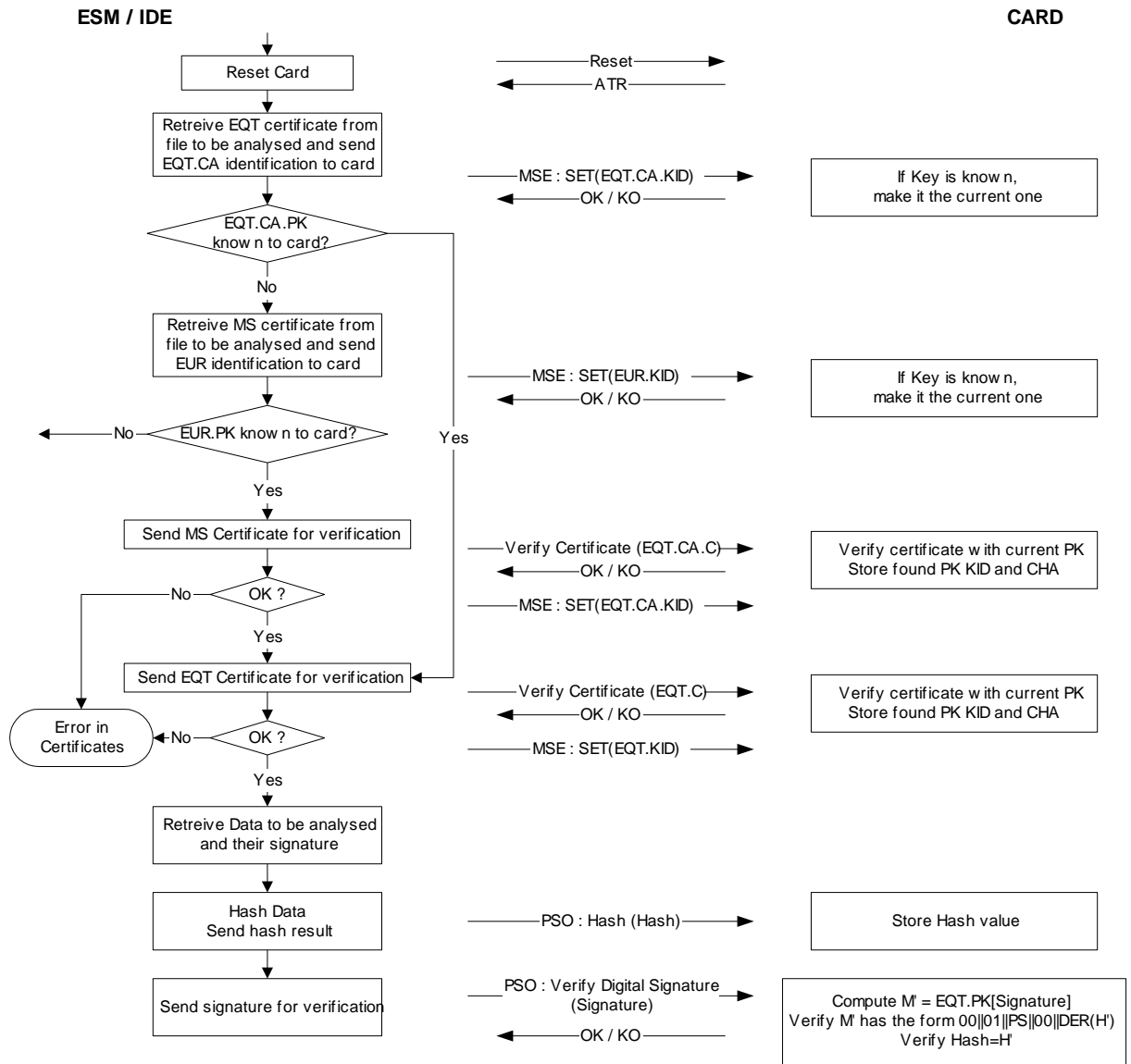
6.2. Signature verification

CSM\_035 Data signature verification on downloaded data shall follow the signature scheme with appendix defined in reference [PKCS1] with the SHA-1 hash function.

The European public key EUR.PK needs to be known independently (and trusted) by the verifier.

The following table illustrates the protocol an IDE carrying a Control card can follow to verify the integrity of data downloaded and stored on the ESM (External Storage media). The control card is used to perform the decipherment of digital signatures. This function may in this case not be implemented in the IDE.

The equipment that has downloaded and signed the data to be analysed is denoted EQT.





## PART B SECOND-GENERATION TACHOGRAPH SYSTEM

### 7. Introduction

#### 7.1. References

The following references are used in this part of this Appendix.

- AES            National Institute of Standards and Technology (NIST), FIPS PUB 197: Advanced Encryption Standard (AES), November 26, 2001
- DSS            National Institute of Standards and Technology (NIST), FIPS PUB 186-4: Digital Signature Standard (DSS), July 2013
- ISO 7816-4    ISO/IEC 7816-4, Identification cards - Integrated circuit cards - Part 4: Organization, security and commands for interchange. Third edition 2013-04-15
- ISO 7816-8    ISO/IEC 7816-8, Identification cards - Integrated circuit cards - Part 8: Commands for security operations. Second edition 2004-06-01
- ISO 8825-1    ISO/IEC 8825-1, Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER). Fourth edition, 2008-12-15
- ISO 9797-1    ISO/IEC 9797-1, Information technology – Security techniques – Message Authentication Codes (MACs) – Part 1: Mechanisms using a block cipher. Second edition, 2011-03-01
- ISO 10116    ISO/IEC 10116, Information technology – Security techniques – Modes of operation of an *n*-bit block cipher. Third edition, 2006-02-01
- ISO 16844-3    ISO/IEC 16844-3, Road vehicles – Tachograph systems – Part 3: Motion sensor interface. First edition 2004, including Technical Corrigendum 1 2006
- RFC 5480    Elliptic Curve Cryptography Subject Public Key Information, March 2009
- RFC 5639    Elliptic Curve Cryptography (ECC) - Brainpool Standard Curves and Curve Generation, 2010
- RFC 5869    HMAC-based Extract-and-Expand Key Derivation Function (HKDF), May 2010

- SHS National Institute of Standards and Technology (NIST), FIPS PUB 180-4: Secure Hash Standard, March 2012
- SP 800-38B National Institute of Standards and Technology (NIST), Special Publication 800-38B: Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, 2005
- TR-03111 BSI Technical Guideline TR-03111, Elliptic Curve Cryptography, version 2.00, 2012-06-28

## 7.2. Notations and Abbreviations

The following notations and abbreviated terms are used in this Appendix:

AES	Advanced Encryption Standard
CA	Certificate Authority
CAR	Certificate Authority Reference
CBC	Cipher Block Chaining (mode of operation)
CH	Command Header
CHA	Certificate Holder Authorisation
CHR	Certificate Holder Reference
CV	Constant Vector
DER	Distinguished Encoding Rules
DO	Data Object
DSRC	Dedicated Short Range Communication
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
ECDH	Elliptic Curve Diffie-Hellman (key agreement algorithm)
EGF	External GNSS Facility
EQT	Equipment
IDE	Intelligent Dedicated Equipment
$K_M$	Motion Sensor Master Key, allowing the pairing of a Vehicle Unit to a Motion Sensor
$K_{M-VU}$	Key inserted in vehicle units, allowing a VU to derive the Motion Sensor Master Key if a workshop card is inserted into the VU
$K_{M-WC}$	Key inserted in workshop cards, allowing a VU to derive the Motion Sensor Master Key if a workshop card is inserted into the VU
MAC	Message Authentication Code
MoS	Motion Sensor
MSB	Most Significant Bit
PKI	Public Key Infrastructure

RCF	Remote Communication Facility
SSC	Send Sequence Counter
SM	Secure Messaging
TDES	Triple Data Encryption Standard
TLV	Tag Length Value
VU	Vehicle Unit
X.C	the public key certificate of user X
X.CA	the certificate authority that issued the certificate of user X
X.CAR	the certificate authority reference mentioned in the certificate of user X
X.CHR	the certificate holder reference mentioned in the certificate of user X
X.PK	public key of user X
X.SK	private key of user X
X.PK <sub>eph</sub>	ephemeral public key of user X
X.SK <sub>eph</sub>	ephemeral private key of user X
'xx'	a hexadecimal value
	concatenation operator

### 7.3. Definitions

The definitions of terms used in this Appendix are included in section I of Annex 1C.

8. Cryptographic systems and algorithms

8.1. Cryptographic Systems

- TCS\_180 Vehicle units and tachograph cards shall use an elliptic curve-based public-key cryptographic system to provide the following security services:
- mutual authentication between a vehicle unit and a card,
  - agreement of AES session keys between a vehicle unit and a card,
  - ensuring the authenticity, integrity and non-repudiation of data downloaded from vehicle units or tachograph cards to external media.
- TCS\_181 Vehicle units and external GNSS facilities shall use an elliptic curve-based public-key cryptographic system to provide the following security services:
- coupling of a vehicle unit and an external GNSS facility,
  - mutual authentication between a vehicle unit and an external GNSS facility,
  - agreement of an AES session key between a vehicle unit and an external GNSS facility.
- TCS\_182 Vehicle units and tachograph cards shall use an AES-based symmetric cryptographic system to provide the following security services:
- ensuring authenticity and integrity of data exchanged between a vehicle unit and a tachograph card,
  - where applicable, ensuring confidentiality of data exchanged between a vehicle unit and a tachograph card.
- TCS\_183 Vehicle units and external GNSS facilities shall use an AES-based symmetric cryptographic system to provide the following security services:
- ensuring authenticity and integrity of data exchanged between a vehicle unit and an external GNSS facility.
- TCS\_184 Vehicle units and motion sensors shall use an AES-based symmetric cryptographic system to provide the following security services:
- pairing of a vehicle unit and a motion sensor,
  - mutual authentication between a vehicle unit and a motion sensor,
  - ensuring confidentiality of data exchanged between a vehicle unit and a motion sensor.
- TCS\_185 Vehicle units and control cards shall use an AES-based symmetric cryptographic system to provide the following security services on the remote communication interface:
- ensuring confidentiality, authenticity and integrity of data transmitted from a vehicle unit to a control card.

Notes:

- Properly speaking, data is transmitted from a vehicle unit to a remote interrogator under the control of a control officer, using a remote communication facility that may be internal or external to the VU, see Appendix 14. However, the remote interrogator sends the received data to a control card for decryption and validation of authenticity. From a security point of view, the remote communication facility and the remote interrogator are fully transparent.
- A workshop card offers the same security services for the DSRC interface as a control card does. This allows a workshop to validate the proper functioning of the remote communication interface of a VU, including security. Please refer to section 9.2.3 for more information.

## 8.2. Cryptographic Algorithms

### 8.2.1 Symmetric Algorithms

TCS\_186 Vehicle units, tachograph cards, motion sensors and external GNSS facilities shall support the AES algorithm as defined in [AES], with key lengths of 128, 192 and 256 bits.

### 8.2.2 Asymmetric Algorithms and Standardized Domain Parameters

TCS\_187 Vehicle units, tachograph cards and external GNSS facilities shall support elliptic curve cryptography with a key size of 256, 384 and 512/521 bits.

TCS\_188 Vehicle units, tachograph cards and external GNSS facilities shall support the ECDSA signing algorithm, as specified in [DSS].

TCS\_189 Vehicle units, tachograph cards and external GNSS facilities shall support the ECKA-EG key agreement algorithm, as specified in [TR 03111].

TCS\_190 Vehicle units, tachograph cards and external GNSS facilities shall support all standardized domain parameters specified in Table 43 below for elliptic curve cryptography.

<i>Name</i>	<i>Size (bits)</i>	<i>Reference</i>	<i>Object identifier</i>
NIST P-256	256	[DSS], [RFC 5480]	secp256r1
BrainpoolP256r1	256	[RFC 5639]	brainpoolP256r1
NIST P-384	384	[DSS], [RFC 5480]	secp384r1
BrainpoolP384r1	384	[RFC 5639]	brainpoolP384r1
BrainpoolP512r1	512	[RFC 5639]	brainpoolP512r1
NIST P-521	521	[DSS], [RFC 5480]	secp521r1

Table 43 Standardized domain parameters

Note: the object identifiers mentioned in the last column of Table 43 are specified in [RFC 5639] for the Brainpool curves and in [RFC 5480] for the NIST curves.

Example 1: the object identifier of the BrainpoolP256r1 curve is {iso(1) identified-organization(3) teletrust(36) algorithm(3) signaturealgorithm(3) ecSign(2) ecStdCurvesAndGeneration(8) ellipticCurve(1) versionOne(1) 7}.

Or in dot notation: 1.3.36.3.3.2.8.1.1.7.

Example 2: the object identifier of the NIST P-384 curve is

{iso(1) identified-organization(3) certicom(132) curve(0) 34}.

Or in dot notation: 1.3.132.0.34.

### 8.2.3 Hashing algorithms

TCS\_191 Vehicle units and tachograph cards shall support the SHA-256, SHA-384 and SHA-512 algorithms specified in [SHS].

### 8.2.4 Cipher Suites

TCS\_192 In case a symmetric algorithm, an asymmetric algorithm and/or a hashing algorithm are used together to form a security protocol, their respective key lengths and hash sizes shall be of (roughly) equal strength. Table 44 shows the allowed cipher suites:

<i>Cipher suite Id</i>	<i>ECC key size (bits)</i>	<i>AES key length (bits)</i>	<i>Hashing algorithm</i>	<i>MAC length (bytes)</i>
CS#1	256	128	SHA-256	8
CS#2	384	192	SHA-384	12
CS#3	512/521	256	SHA-512	16

**Table 44 Allowed cipher suites**

Note: ECC keys sizes of 512 bits and 521 bits are considered to be equal in strength for all purposes within this Appendix.

## 9. Keys and Certificates

### 9.1. Asymmetric Key Pairs and Public Key Certificates

#### 9.1.1 General

Note: the keys described in this section are used for mutual authentication and secure messaging between vehicle units and tachograph cards and between vehicle units and external GNSS facilities. These processes are described in detail in chapters 10 and 11 of this Appendix.

TCS\_193 Within the European Smart Tachograph system, ECC key pairs and corresponding certificates shall be generated and managed through three functional hierarchical levels:

- European level,
- Member State level,
- Equipment level.

TCS\_194 Within the entire European Smart Tachograph system, public and private keys and certificates shall be generated, managed and communicated using standardized and secure methods.

#### 9.1.2 European Level

TCS\_195 At European level, a single unique ECC key pair designated as EUR shall be generated. It shall consist of a private key (EUR.SK) and a public key (EUR.PK). This key pair shall form the root key pair of the entire European Smart Tachograph PKI. This task shall be handled by a European Root Certificate Authority (ERCA), under the authority and responsibility of the European Commission.

TCS\_196 The ERCA shall use the European private key to sign a (self-signed) root certificate of the European public key, and shall communicate this European root certificate to all Member States.

TCS\_197 The ERCA shall use the European private key to sign the certificates of the Member States public keys upon request. The ERCA shall keep records of all signed Member State public key certificates.

TCS\_198 As shown in Figure 1 in section 9.1.7, the ERCA shall generate a new European root key pair every 17 years. Whenever the ERCA generates a new European root key pair, it shall create a new self-signed root certificate for the new European public key. The validity period of a European root certificate shall be 34 years plus 3 months.

Note: The introduction of a new root key pair also implies that ERCA will generate a new motion sensor master key and a new DSRC master key, see sections 9.2.2.2 and 9.2.3.2.

TCS\_199 Before generating a new European root key pair, the ERCA shall conduct an analysis of the cryptographic strength that is needed for the new key pair, given it should stay secure for the next 34 years. If found necessary, the ERCA shall switch to a cipher suite that is stronger than the current one, as specified in TCS\_192.

TCS\_200 Whenever it generates a new European root key pair, the ERCA shall create a link certificate for the new European public key and sign it with the previous European private key. The validity period of the link certificate shall be 17 years. This is shown in Figure 1 in section 9.1.7 as well.

Note: Since a link certificate contains the ERCA generation  $X$  public key and is signed with the ERCA generation  $X-1$  private key, a link certificate offers equipment issued under generation  $X-1$  a method to trust equipment issued under generation  $X$ .

TCS\_201 The ERCA shall not use the private key of a root key pair for any purpose after the moment a new root key certificate becomes valid.

TCS\_202 At any moment in time, the ERCA shall dispose of the following cryptographic keys and certificates:

- The current EUR key pair and corresponding certificate
- All previous EUR certificates to be used for the verification of MSCA certificates that are still valid

- Link certificates for all generations of EUR certificates except the first one

### 9.1.3 Member State Level

- TCS\_203 At Member State level, all Member States required to sign tachograph card certificates shall generate one or more unique ECC key pairs designated as MSCA\_Card. All Member States required to sign certificates for vehicle units or external GNSS facilities shall additionally generate one or more unique ECC key pairs designated as MSCA\_VU-EGF.
- TCS\_204 The task of generating Member State key pairs shall be handled by a Member State Certificate Authority (MSCA). Whenever a MSCA generates a Member State key pair, it shall send the public key to the ERCA in order to obtain a corresponding Member State certificate signed by the ERCA.
- TCS\_205 An MSCA shall choose the strength of a Member State key pair equal to the strength of the European root key pair used to sign the corresponding Member State certificate.
- TCS\_206 An MSCA\_VU-EGF key pair, if present, shall consist of private key MSCA\_VU-EGF.SK and public key MSCA\_VU-EGF.PK. An MSCA shall use the MSCA\_VU-EGF.SK private key exclusively to sign the public key certificates of vehicle units and external GNSS facilities.
- TCS\_207 An MSCA\_Card key pair shall consist of private key MSCA\_Card.SK and public key MSCA\_Card.PK. An MSCA shall use the MSCA\_Card.SK private key exclusively to sign the public key certificates of tachograph cards.
- TCS\_208 An MSCA shall keep records of all signed VU certificates, external GNSS facility certificates and card certificates, together with the identification of the equipment for which each certificate is intended.
- TCS\_209 The validity period of an MSCA\_VU-EGF certificate shall be 17 years plus 3 months. The validity period of an MSCA\_Card certificate shall be 7 years plus 1 month.
- TCS\_210 As shown in Figure 1 in section 9.1.7, the private key of a MSCA\_VU-EGF key pair and the private key of a MSCA\_Card key pair shall have a key usage period of two years.
- TCS\_211 An MSCA shall not use the private key of an MSCA\_VU-EGF key pair for any purpose after the moment its usage period has ended. Neither shall an MSCA use the private key of an MSCA\_Card key pair for any purpose after the moment its usage period has ended.
- TCS\_212 At any moment in time, an MSCA shall dispose of the following cryptographic keys and certificates:
- The current MSCA\_Card key pair and corresponding certificate
  - All previous MSCA\_Card certificates to be used for the verification of the certificates of tachograph cards that are still valid
  - The current EUR certificate necessary for the verification of the current MSCA certificate



- All previous EUR certificates necessary for the verification of all MSCA certificates that are still valid

- TCS\_213 If an MSCA is required to sign certificates for vehicle units or external GNSS facilities, it shall additionally dispose of the following keys and certificates:
- The current MSCA\_VU-EGF key pair and corresponding certificate
  - All previous MSCA\_VU-EGF public keys to be used for the verification of the certificates of VUs or external GNSS facilities that are still valid

#### 9.1.4 Equipment Level: Vehicle Units

- TCS\_214 Two unique ECC key pairs shall be generated for each vehicle unit, designated as VU\_MA and VU\_Sign. This task is handled by VU manufacturers. Whenever a VU key pair is generated, the party generating the key shall send the public key to the MSCA of the country in which it resides, in order to obtain a corresponding VU certificate signed by the MSCA. The private key shall be used only by the vehicle unit.
- TCS\_215 The VU\_MA and VU\_Sign certificates of a given vehicle unit shall have the same Certificate Effective Date.
- TCS\_216 A VU manufacturer shall choose the strength of a VU key pair equal to the strength of the MSCA key pair used to sign the corresponding VU certificate.
- TCS\_217 A vehicle unit shall use its VU\_MA key pair, consisting of private key VU\_MA.SK and public key VU\_MA.PK, exclusively to perform VU Authentication towards tachograph cards and external GNSS facilities, as specified in sections 10.3 and 11.4 of this Appendix.
- TCS\_218 A vehicle unit shall be capable of generating ephemeral ECC key pairs and shall use an ephemeral key pair exclusively to perform session key agreement with a tachograph card or external GNSS facility, as specified in sections 10.4 and 11.4 of this Appendix.
- TCS\_219 A vehicle unit shall use the private key VU\_Sign.SK of its VU\_Sign key pair exclusively to sign downloaded data files, as specified in chapter 14 of this Appendix. The corresponding public key VU\_Sign.PK shall be used exclusively to verify signatures created by the vehicle unit.
- TCS\_220 As shown in Figure 1 in section 9.1.7, the validity period of a VU\_MA certificate shall be 15 years and 3 months. The validity period of a VU\_Sign certificate shall also be 15 years and 3 months.

#### Notes:

- The extended validity period of a VU\_Sign certificate allows a Vehicle Unit to create valid signatures over downloaded data during the first three months after it has expired, as required in Regulation (EU) N°. 581/2010.
- The extended validity period of a VU\_MA certificate is needed to allow the VU to authenticate to a control card or a company card during the first three months after it has expired, such that it is possible to perform a data download.

TCS\_221 A vehicle unit shall not use the private key of a VU key pair for any purpose after the corresponding certificate has expired.

TCS\_222 The VU key pairs (except ephemeral keys pairs) and corresponding certificates of a given vehicle unit shall not be replaced or renewed in the field once the vehicle unit has been put in operation.

Notes:

- Ephemeral key pairs are not included in this requirement, as a new ephemeral key pair is generated by a VU each time Chip Authentication and session key agreement is performed, see section 10.4. Note that ephemeral key pairs do not have corresponding certificates.
- This requirement does not forbid the possibility of replacing static VU key pairs during a refurbishment or repair in a secure environment controlled by the VU manufacturer.

TCS\_223 When put in operation, vehicle units shall contain the following cryptographic keys and certificates:

- The VU\_MA private key and corresponding certificate
- The VU\_Sign private key and corresponding certificate
- The MSCA\_VU-EGF certificate containing the MSCA\_VU-EGF.PK public key to be used for verification of the VU\_MA certificate and VU\_Sign certificate
- The EUR certificate containing the EUR.PK public key to be used for verification of the MSCA\_VU-EGF certificate
- The EUR certificate whose validity period directly precedes the validity period of the EUR certificate to be used to verify the MSCA\_VU-EGF certificate, if existing
- The link certificate linking these two EUR certificates, if existing

TCS\_224 In addition to the cryptographic keys and certificates listed in TCS\_223, vehicle units shall also contain the keys and certificates specified in Part A of this Appendix, allowing a vehicle unit to interact with first-generation tachograph cards.

9.1.5 Equipment Level: Tachograph Cards

TCS\_225 One unique ECC key pair, designated as Card\_MA, shall be generated for each tachograph card. A second unique ECC key pair, designated as Card\_Sign, shall additionally be generated for each driver card and each workshop card. This task may be handled by card manufacturers or card personalisers. Whenever a card key pair is generated, the party generating the key shall send the public key to the MSCA of the country in which it resides, in order to obtain a corresponding card certificate signed by the MSCA. The private key shall be used only by the tachograph card.

TCS\_226 The Card\_MA and Card\_Sign certificates of a given driver card or workshop card shall have the same Certificate Effective Date.

TCS\_227 A card manufacturer or card personaliser shall choose the strength of a card key pair equal to the strength of the MSCA key pair used to sign the corresponding card certificate.

TCS\_228 A tachograph card shall use its Card\_MA key pair, consisting of private key Card\_MA.SK and public key Card\_MA.PK, exclusively to perform mutual authentication and session key agreement towards vehicle units, as specified in sections 10.3 and 10.4 of this Appendix.

TCS\_229 A driver card or workshop card shall use the private key Card\_Sign.SK of its Card\_Sign key pair exclusively to sign downloaded data files, as specified in chapter 14 of this Appendix. The corresponding public key Card\_Sign.PK shall be used exclusively to verify signatures created by the card.

TCS\_230 The validity period of a Card\_MA certificate shall be as follows:

- For driver cards: 5 years
- For company cards: 2 years
- For control cards: 2 years
- For workshop cards: 1 year

TCS\_231 The validity period of a Card\_Sign certificate shall be as follows:

- For driver cards: 5 years and 1 month
- For workshop cards: 1 year and 1 month

Note: the extended validity period of a Card\_Sign certificate allows a driver card to create valid signatures over downloaded data during the first month after it has expired. This is necessary in view of Regulation (EU) N°. 581/2010, which requires that a data download from a driver card must be possible up to 28 days after the last data has been recorded.

TCS\_232 The key pairs and corresponding certificates of a given tachograph card shall not be replaced or renewed once the card has been issued.

TCS\_233 When issued, tachograph cards shall contain the following cryptographic keys and certificates:

- The Card\_MA private key and corresponding certificate
- For driver cards and workshop cards additionally: the Card\_Sign private key and corresponding certificate
- The MSCA\_Card certificate containing the MSCA\_Card.PK public key to be used for verification of the Card\_MA certificate and Card\_Sign certificate
- The EUR certificate containing the EUR.PK public key to be used for verification of the MSCA\_Card certificate.
- The EUR certificate whose validity period directly precedes the validity period of the EUR certificate to be used to verify the MSCA\_Card certificate, if existing.

- The link certificate linking these two EUR certificates, if existing.

TCS\_234 In addition to the cryptographic keys and certificates listed in TCS\_233, tachograph cards shall also contain the keys and certificates specified in Part A of this Appendix, allowing these cards to interact with first-generation VUs.

#### 9.1.6 Equipment Level: External GNSS Facilities

TCS\_235 One unique ECC key pair shall be generated for each external GNSS facility, designated as EGF\_MA. This task is handled by external GNSS facility manufacturers. Whenever an EGF\_MA key pair is generated, the public key shall be sent to the MSCA of the country in which it resides, in order to obtain a corresponding EGF\_MA certificate signed by the MSCA. The private key shall be used only by the external GNSS facility.

TCS\_236 An EGF manufacturer shall choose the strength of an EGF\_MA key pair equal to the strength of the MSCA key pair used to sign the corresponding EGF\_MA certificate.

TCS\_237 An external GNSS facility shall use its EGF\_MA key pair, consisting of private key EGF\_MA.SK and public key EGF\_MA.PK, exclusively to perform mutual authentication and session key agreement towards vehicle units, as specified in section 11.4 and 11.4 of this Appendix.

TCS\_238 The validity period of an EGF\_MA certificate shall be 15 years.

TCS\_239 An external GNSS facility shall not use the private key of its EGF\_MA key pair for coupling to a vehicle unit after the corresponding certificate has expired.

Note: as explained in section 11.3.3, an EGF may potentially use its private key for mutual authentication towards the VU it is already coupled to, even after the corresponding certificate has expired.

TCS\_240 The EGF\_MA key pair and corresponding certificate of a given external GNSS facility shall not be replaced or renewed in the field once the EGF has been put in operation.

Note: This requirement does not forbid the possibility of replacing EGF key pairs during a refurbishment or repair in a secure environment controlled by the EGF manufacturer.

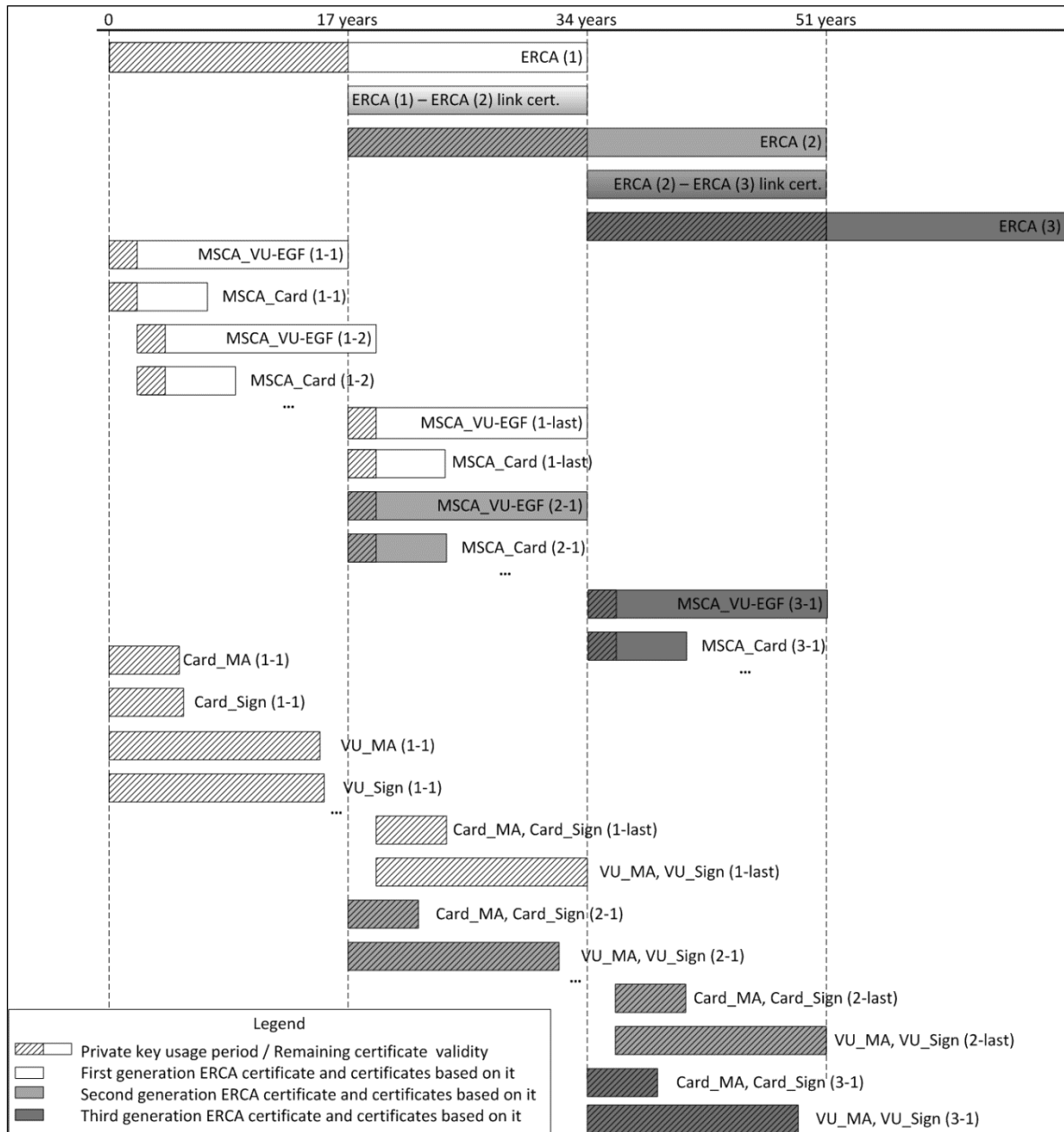
TCS\_241 When put in operation, an external GNSS facility shall contain the following cryptographic keys and certificates:

- The EGF\_MA private key and corresponding certificate
- The MSCA\_VU-EGF certificate containing the MSCA\_VU-EGF.PK public key to be used for verification of the EGF\_MA certificate
- The EUR certificate containing the EUR.PK public key to be used for verification of the MSCA\_VU-EGF certificate

- The EUR certificate whose validity period directly precedes the validity period of the EUR certificate to be used to verify the MSCA\_VU-EGF certificate, if existing
- The link certificate linking these two EUR certificates, if existing

9.1.7 Overview: Certificate Replacement

Figure 1 below shows how different generations of ERCA root certificates, ERCA link certificates, MSCA certificates and equipment (VU and card) certificates are issued and used over time:



**Figure 1 Issuance and usage of different generations of ERCA root certificates, ERCA link certificates, MSCA certificates and equipment certificates**

Notes to Figure 1:

1. Different generations of the root certificate are indicated by a number in brackets. E.g. ERCA (1) is the first generation of ERCA root certificate; ERCA (2) is the second generation, etc.
2. Other certificates are indicated by two numbers in brackets, the first one indicating the root certificate generation under which they are issued, the second one the generation of the certificate itself. E.g.

MSCA\_Card (1-1) is the first MSCA\_Card certificate issued under ERCA (1); MSCA\_Card (2-1) is the first MSCA\_Card certificate issued under ERCA (2); MSCA\_Card (2-last) is the last MSCA\_Card certificate issued under ERCA (2); Card\_MA(2-1) is the first Card certificate for mutual authentication that is issued under ERCA (2), etc.

3. The MSCA\_Card (2-1) and MSCA\_Card (1-last) certificates are issued at almost but not exactly the same date. MSCA\_Card (2-1) is the first MSCA\_Card certificate issued under ERCA (2) and will be issued slightly later than MSCA\_Card (1-last), the last MSCA\_Card certificate under ERCA (1).
4. As shown in the figure, the first VU and Card certificates issued under ERCA (2) will appear almost two years before the last VU and Card certificates issued under ERCA (1) will appear. This is because of the fact that VU and Card certificates are issued under an MSCA certificate, not directly under the ERCA certificate. The MSCA (2-1) certificate will be issued directly after ERCA (2) becomes valid, but the MSCA (1-last) certificate will be issued only slightly before that time, at the last moment the ERCA (1) certificate is still valid. Therefore, these two MSCA certificates will have almost the same validity period, despite the fact that they are of different generations.
5. The validity period shown for cards is the one for driver cards (5 years).
6. To save space, the difference in validity period between the Card\_MA and Card\_Sign certificates and between the VU\_MA and VU\_Sign certificates is shown only for the first generation.

## 9.2. Symmetric Keys

### 9.2.1 Keys for Securing VU – Motion Sensor Communication

### 9.2.2 Keys for Securing DSRC Communication

#### 9.2.2.1 General

Note: readers of this section are supposed to be familiar with the contents of [ISO 16844-3] describing the interface between a vehicle unit and a motion sensor. The pairing process between a VU and a motion sensor is described in detail in chapter 12 of this Appendix.

A number of symmetric keys is needed for pairing vehicle units and motion sensors, for mutual authentication between vehicle units and motion sensors and for encrypting communication between vehicle units and motion sensors, as shown in Table 45. All of these keys shall be AES keys, with a key length equal to the length of the motion sensor master key, which shall be linked to the length of the (foreseen) European root key pair as described in TCS\_192.

<i>Key</i>	<i>Symbol</i>	<i>Generated by</i>	<i>Generation method</i>	<i>Stored by</i>
Motion Sensor Master Key – VU part	$K_{M-VU}$	ERCA	Random	ERCA, MSCAs involved in issuing VUs certificates, VU manufacturers, vehicle units
Motion Sensor Master Key – Workshop part	$K_{M-WC}$	ERCA	Random	ERCA, MSCAs, card manufacturers, workshop cards

<i>Key</i>	<i>Symbol</i>	<i>Generated by</i>	<i>Generation method</i>	<i>Stored by</i>
Motion Sensor Master Key	$K_M$	Not independently generated	Calculated as $K_M = K_{M-VU} \text{ XOR } K_{M-WC}$	ERCA, MSCAs involved in issuing motion sensors keys (optionally)*
Identification Key	$K_{ID}$	Not independently generated	Calculated as $K_{ID} = K_M \text{ XOR } CV$ , where CV is specified in TCS_248	ERCA, MSCAs involved in issuing motion sensors keys (optionally)*
Pairing Key	$K_P$	Motion sensor manufacturer	Random	One motion sensor
Session Key	$K_S$	VU (during pairing of VU and motion sensor)	Random	One VU and one motion sensor

**Table 45 Keys for securing vehicle unit - motion sensor communication**

\*Storage of  $K_M$  and  $K_{ID}$  is optional, as these keys can be derived from  $K_{M-VU}$ ,  $K_{M-WC}$  and CV.

TCS\_243 The European Root Certificate Authority shall generate  $K_{M-VU}$  and  $K_{M-WC}$ , two random and unique AES keys from which the motion sensor master key  $K_M$  can be calculated as  $K_M = K_{M-VU} \text{ XOR } K_{M-WC}$ . The ERCA shall communicate  $K_M$ ,  $K_{M-VU}$  and  $K_{M-WC}$  to Member State Certificate Authorities upon their request.

TCS\_244 The ERCA shall assign to each motion sensor master key  $K_M$  a unique version number, which shall also be applicable for the constituting keys  $K_{M-VU}$  and  $K_{M-WC}$  and for the related identification key  $K_{ID}$ . The ERCA shall inform the MSCAs about the version number when sending  $K_{M-VU}$  and  $K_{M-WC}$  to them.

Note: The version number is used to distinguish different generations of these keys, as explained in detail in section 9.2.2.2.

TCS\_245 A Member State Certificate Authority shall forward  $K_{M-VU}$ , together with its version number, to vehicle unit manufacturers upon their request. The VU manufacturers shall insert  $K_{M-VU}$  and its version number in all manufactured VUs.

TCS\_246 A Member State Certificate Authority shall ensure that  $K_{M-WC}$ , together with its version number, is inserted in every workshop card issued under its responsibility.

Notes:

- See the description of data type SensorInstallationSecData in Appendix 2.



- as explained in section 9.2.2.2, in fact multiple generations of  $K_{M-WC}$  may have to be inserted in a single workshop card.

TCS\_247 In addition to the AES key specified in TCS\_246, a MSCA shall ensure that the TDES key  $K_{mWC}$ , specified in requirement CSM\_037 in Part A of this Appendix, is inserted in every workshop card issued under its responsibility.

Notes:

- This allows a second-generation workshop card to be used for coupling a first-generation VU.
- A second-generation workshop card will contain two different applications, one complying with Part B of this Appendix and one complying with Part A. The latter will contain the TDES key  $K_{mWC}$ .

TCS\_248 An MSCA involved in issuing motion sensors shall derive the identification key from the motion sensor master key by XORing it with a constant vector CV. The value of CV shall be as follows:

- For 128-bit motion sensor master keys: CV = 'B6 44 2C 45 0E F8 D3 62 0B 7A 8A 97 91 E4 5E 83'
- For 192-bit motion sensor master keys: CV = '72 AD EA FA 00 BB F4 EE F4 99 15 70 5B 7E EE BB 1C 54 ED 46 8B 0E F8 25'
- For 256-bit motion sensor master keys: CV = '1D 74 DB F0 34 C7 37 2F 65 55 DE D5 DC D1 9A C3 23 D6 A6 25 64 CD BE 2D 42 0D 85 D2 32 63 AD 60'

Note: the constant vectors have been generated as follows:

Pi\_10 = first 10 bytes of the decimal portion of the mathematical constant  $\pi$  = '24 3F 6A 88 85 A3 08 D3 13 19'

CV\_128-bits = first 16 bytes of SHA-256(Pi\_10)

CV\_192-bits = first 24 bytes of SHA-384(Pi\_10)

CV\_256-bits = first 32 bytes of SHA-512(Pi\_10)

TCS\_249 Motion sensor manufacturers shall generate a random and unique pairing key  $K_P$  for every motion sensor, and shall send each pairing key to a Member State Certificate Authority. The MSCA shall encrypt each pairing key separately with the motion sensor master key  $K_M$  and shall return the encrypted key to the motion sensor manufacturer. For each encrypted key, the MSCA shall notify the motion sensor manufacturer of the version number of the associated  $K_M$ .

Note: as explained in section 9.2.2.2, in fact a motion sensor manufacturer may have to generate multiple unique pairing keys for a single motion sensor.

TCS\_250 Motion sensor manufacturers shall generate a unique serial number for every motion sensor, and shall send all serial numbers to a Member State Certificate Authority. The MSCA shall encrypt each serial number separately with the identification key  $K_{ID}$  and shall return the encrypted serial number to the motion sensor manufacturer. For each encrypted serial number, the MSCA shall notify the motion sensor manufacturer of the version number of the associated  $K_{ID}$ .

TCS\_251 For requirements TCS\_249 and TCS\_250, the MSCA shall use the AES algorithm in the Cipher Block Chaining mode of operation, as defined in [ISO 10116], with an interleave parameter  $m = 1$  and an initialization vector  $SV = '00' \{16\}$ , i.e. sixteen bytes with binary value 0. When necessary, the MSCA shall use padding method 2 defined in [ISO 9797-1].

TCS\_252 The motion sensor manufacturer shall store the encrypted pairing key and the encrypted serial number in the intended motion sensor, together with the corresponding plain text values and the version number of  $K_M$  and  $K_{ID}$  used for encrypting.

Note: as explained in section 9.2.2.2, in fact a motion sensor manufacturer may have to insert multiple encrypted pairing keys and multiple encrypted serial numbers in a single motion sensor.

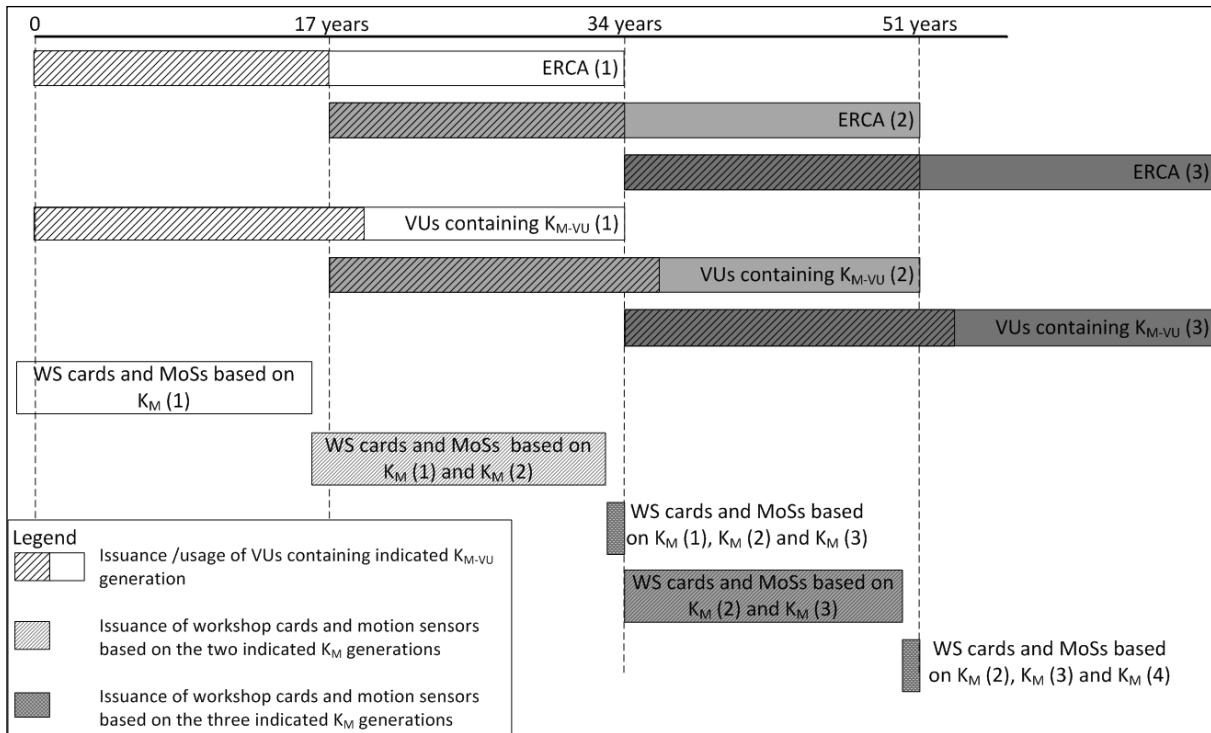
TCS\_253 In addition to the AES-based cryptographic material specified in TCS\_252, a motion sensor manufacturer may also store in each motion sensor the TDES-based cryptographic material specified in requirement CSM\_037 in Part A of this Appendix.

Note: doing so will allow a second-generation motion sensor to be coupled to a first-generation VU.

TCS\_254 The length of the session key  $K_S$  generated by a VU during the pairing to a motion sensor shall be linked to the length of its  $K_{M-VU}$ , as described in TCS\_192.

#### 9.2.2.2 Motion Sensor Master Key Replacement in Second-Generation Equipment

TCS\_255 Each motion sensor master key and all related keys (see Table 45) is associated to a particular generation of the ERCA root key pair. These keys shall therefore be replaced every 17 years. The validity period of each motion sensor master key generation shall begin one year before the associated ERCA root key pair becomes valid and shall end when the associated ERCA root key pair expires. This is depicted in Figure 2.



**Figure 2 Issuance and usage of different generations of the motion sensor master key in vehicle units, motions sensors and workshop cards**

TCS\_256 At least one year before generating a new European root key pair, as described in TCS\_198, the ERCA shall generate a new motion sensor master key  $K_M$  by generating a new  $K_{M-VU}$  and  $K_{M-WC}$ . The length of the motion sensor master key shall be linked to the foreseen strength of the new European root key pair, according to TCS\_192. The ERCA shall communicate the new  $K_M$ ,  $K_{M-VU}$  and  $K_{M-WC}$  to the MSCAs upon their request, together with their version number.

TCS\_257 An MSCA shall ensure that all valid generations of  $K_{M-WC}$  are stored in every workshop card issued under its authority, together with their version numbers, as shown in Figure 2.

Note: this implies that in the last year of the validity period of an ERCA certificate, workshop cards will be issued with three different generations of  $K_{M-WC}$ , as shown in Figure 2.

TCS\_258 In relation to the process described in TCS\_249 and TCS\_250 above: An MSCA shall encrypt each pairing key  $K_P$  it receives from a motion sensor manufacturer separately with each valid generation of the motion sensor master key  $K_M$ . An MSCA shall also encrypt each serial number it receives from a motion sensor manufacturer separately with each valid generation of the identification key  $K_{ID}$ . A motion sensor manufacturer shall store all encryptions of the pairing key and all encryptions of the serial number in the intended motion sensor, together with the corresponding plain text values and the version number(s) of  $K_M$  and  $K_{ID}$  used for encrypting.

Note: This implies that in the last year of the validity period of an ERCA certificate, motion sensors will be issued with encrypted data based on three different generations of  $K_M$ , as shown in Figure 2.

TCS\_259 In relation to the process described in TCS\_249 above: Since the length of the pairing key  $K_P$  shall be linked to the length of  $K_M$  (see TCS\_242), a motion sensor manufacturer may have to generate up to three different pairing keys (of different lengths) for one motion sensor, in case subsequent generations of  $K_M$  have different lengths. In such a case, the manufacturer shall send each pairing key to the MSCA. The MSCA shall ensure that each pairing key is encrypted with the correct generation of the motion sensor master key, i.e. the one having the same length.

Note: In case the motion sensor manufacturer chooses to generate a TDES-based pairing key for a second-generation motion sensor (see TCS\_253), the manufacturer shall indicate to the MSCA that the TDES-based motion sensor master key must be used for encrypting this pairing key. This is because the length of a TDES key may be equal to that of an AES key, so the MSCA cannot judge from the key length alone.

TCS\_260 Vehicle unit manufacturers shall insert only one generation of  $K_{M-VU}$  in each vehicle unit, together with its version number. This  $K_{M-VU}$  generation shall be linked to the ERCA certificate upon which the VU's certificates are based.

Notes:

- A vehicle unit based on the generation  $X$  ERCA certificate shall only contain the generation  $X$   $K_{M-VU}$ , even if it is issued after the start of the validity period of the generation  $X+1$  ERCA certificate. This is shown in Figure 2.
- A VU of generation  $X$  cannot be paired to a motion sensor of generation  $X-1$ .
- Since workshop cards have a validity period of one year, the result of TCS\_255 - TCS\_260 is that all workshop cards will contain the new  $K_{M-WC}$  at the moment the first VU containing the new  $K_{M-VU}$  is issued. Therefore, such a VU will always be able to calculate the new  $K_M$ . Moreover, by that time most new motion sensors will contain encrypted data based on the new  $K_M$  as well.

### 9.2.3 Keys for Securing DSRC Communication

#### 9.2.3.1 General

TCS\_261 The authenticity and confidentiality of data communicated from a vehicle unit to a control authority over a DSRC remote communication channel shall be ensured by means of a set of VU-specific AES keys derived from a single DSRC master key,  $K_{M-DSRC}$ .

TCS\_262 The DSRC master key  $K_{M-DSRC}$  shall be an AES key that is securely generated, stored and distributed by the ERCA. The key length may be 128, 192 or 256 bits and shall be linked to the length of the European root key pair, as described in TCS\_192.

TCS\_263 The ERCA shall communicate the DSRC master key to Member State Certificate Authorities upon their request in a secure manner, to allow them to derive VU-specific DSRC keys and to ensure that the DSRC master key is inserted in all control cards and workshop cards issued under their responsibility.

TCS\_264 The ERCA shall assign to each DSRC master key a unique version number. The ERCA shall inform the MSCAs about the version number when sending the DSRC master key to them.

Note: The version number is used to distinguish different generations of the DSRC master key, as explained in detail in section 9.2.3.2.

TCS\_265 For every vehicle unit, the vehicle unit manufacturer shall create a unique VU serial number and shall send this number to its Member State Certificate Authority in a request to obtain a set of two VU-specific DSRC keys. The VU serial number shall have data type `VuSerialNumber`, and the Distinguished Encoding Rules (DER) according to [ISO 8825-1] shall be used for encoding.

TCS\_266 Upon receiving a request for VU-specific DSRC keys, the MSCA shall derive two AES keys for the vehicle unit, called `K_VUDSRC_ENC` and `K_VUDSRC_MAC`. These VU-specific keys shall have the same length as the DSRC master key. The MSCA shall use the key derivation function defined in [RFC 5869]. The hash function that is necessary to instantiate the HMAC-Hash function shall be linked to the length of the DSRC master key, as described in TCS\_192. The key derivation function in [RFC 5869] shall be used as follows:

Step 1 (Extract):

- $PRK = \text{HMAC-Hash}(salt, IKM)$  where *salt* is an empty string ‘’ and *IKM* is  $K_{DSRC}$ .

Step 2 (Expand):

- $OKM = T(1)$ , where
  - $T(1) = \text{HMAC-Hash}(PRK, T(0) \parallel info \parallel '01')$  with
    - o  $T(0) =$  an empty string (‘’)
    - o *info* = VU serial number as specified in TCS\_265
- $K_{VU_{DSRC}_{ENC}} =$  first *L* octets of *OKM* and  
 $K_{VU_{DSRC}_{MAC}} =$  last *L* octets of *OKM*  
 where *L* is the required length of  $K_{VU_{DSRC}_{ENC}}$  and  $K_{VU_{DSRC}_{MAC}}$  in octets.

TCS\_267 The MSCA shall distribute `K_VUDSRC_ENC` and `K_VUDSRC_MAC` to the VU manufacturer in a secure manner for insertion in the intended vehicle unit.

TCS\_268 When issued, a vehicle unit shall have stored `K_VUDSRC_ENC` and `K_VUDSRC_MAC` in its secure memory, in order to be able to ensure the integrity, authenticity and confidentiality of data sent over the remote communication channel. A vehicle unit shall also store the version number of the DSRC master key used to derive these VU-specific keys.

TCS\_269 When issued, control cards and workshop cards shall have stored  $K_{DSRC}$  in their secure memory, in order to be able to verify the integrity and authenticity of data sent by a VU over the remote communication channel and to decrypt this data. Control cards and workshop cards shall also store the version number of the DSRC master key.

Note: as explained in section 9.2.3.2, in fact multiple generations of  $K_{DSRC}$  may have to be inserted in a single workshop card or control card.

TCS\_270 The MSCA shall keep records of all VU-specific DSRC keys it generated, their version number and the identification of the VU for which each set of keys is intended.

9.2.3.2 DSRC Master Key Replacement

TCS\_271 Each DSRC master key is associated to a particular generation of the ERCA root key pair. The ERCA shall therefore replace the DSRC master key every 17 years. The validity period of each DSRC master key generation shall begin two years before the associated ERCA root key pair becomes valid and shall end when the associated ERCA root key pair expires. This is depicted in Figure 3.

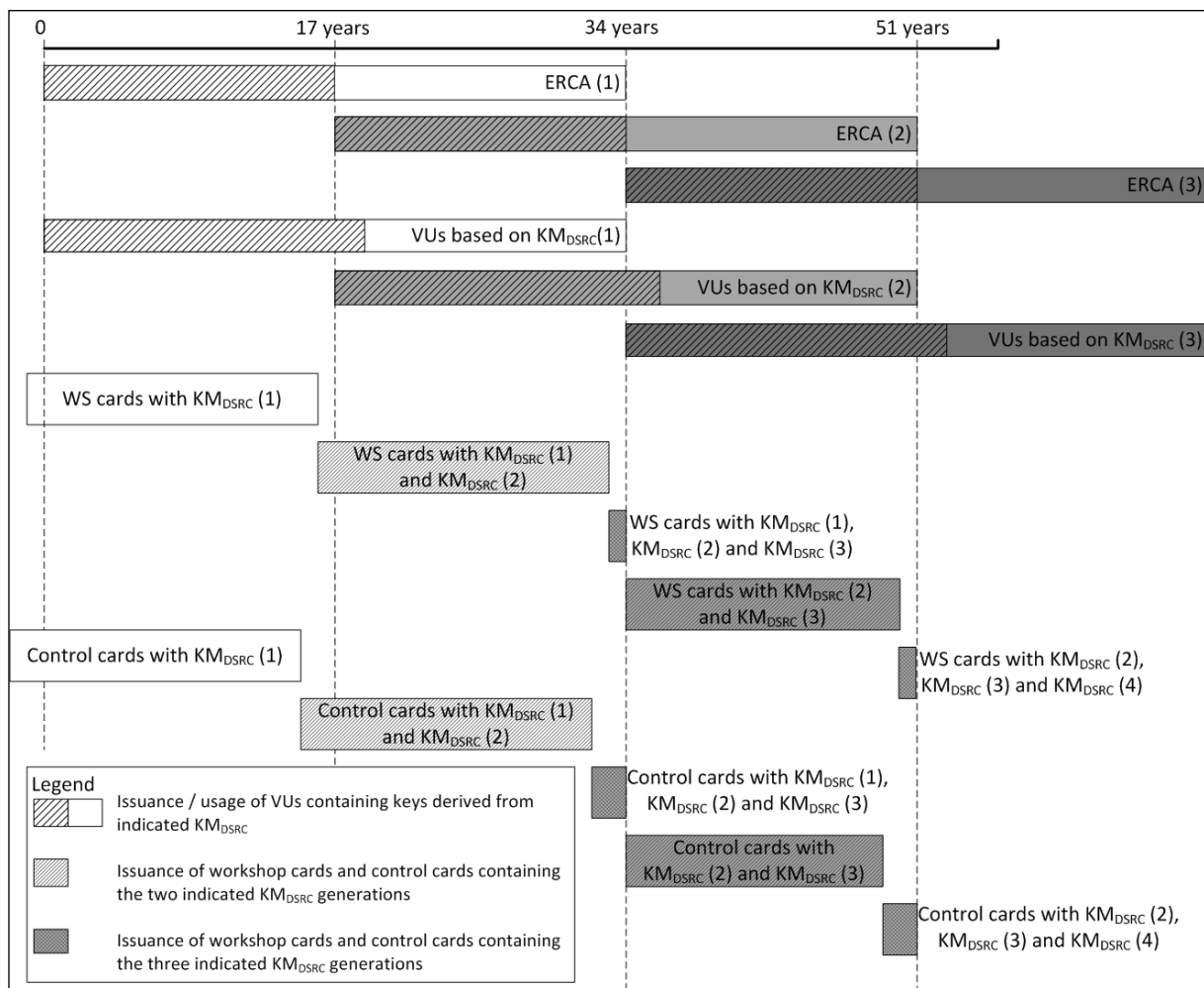


Figure 3 Issuance and usage of different generations of the DSRC master key in vehicle units, workshop cards and control cards

TCS\_272 At least two years before generating a new European root key pair, as described in TCS\_198, the ERCA shall generate a new DSRC master key. The length of the DSRC key shall be linked to the foreseen strength of the new European root key pair, according to TCS\_192. The ERCA shall communicate the new DSRC master key to the MSCAs upon their request, together with its version number.

TCS\_273 An MSCA shall ensure that all valid generations of  $KM_{DSRC}$  are stored in every control card issued under its authority, together with their version numbers, as shown in Figure 3.

Note: this implies that in the last two years of the validity period of an ERCA certificate, control cards will be issued with three different generations of  $KM_{DSRC}$ , as shown in Figure 3.

TCS\_274 An MSCA shall ensure that all generations of  $KM_{DSRC}$  that have been valid for at least a year and are still valid, are stored in every workshop card issued under its authority, together with their version numbers, as shown in Figure 3.

Note: this implies that in the last year of the validity period of an ERCA certificate, workshop cards will be issued with three different generations of  $KM_{DSRC}$ , as shown in Figure 3.

TCS\_275 Vehicle unit manufacturers shall insert only one set of VU-specific DSRC keys into each vehicle unit, together with its version number. This set of keys shall be derived from the  $KM_{DSRC}$  generation linked to the ERCA certificate upon which the VU's certificates are based.

Notes:

- This implies that a vehicle unit based on the generation  $X$  ERCA certificate shall only contain the generation  $X$   $K_{VU_{DSRC\_ENC}}$  and  $K_{VU_{DSRC\_MAC}}$ , even if the VU is issued after the start of the validity period of the generation  $X+1$  ERCA certificate. This is shown in Figure 3.
- Since workshop cards have a validity period of one year and control cards of two years, the result of TCS\_273 - TCS\_275 is that all workshop cards and control cards will contain the new DSRC master key at the moment the first VU containing VU-specific keys based on that master key will be issued.

### 9.3. Certificates

#### 9.3.1 General

TCS\_276 All certificates in the European Smart Tachograph system shall be self-descriptive, card-verifiable (CV) certificates according to [ISO 7816-4] and [ISO 7816-8].

TCS\_277 The Distinguished Encoding Rules (DER) according to [ISO 8825-1] shall be used to encode both ASN.1 data structures and (application specific) data objects within certificates.

Note: this encoding results in a Tag-Length-Value (TLV) structure as follows:

- Tag: The tag is encoded in one or two octets and indicates the content.

Length: The length is encoded as an unsigned integer in one, two, or three octets, resulting in a maximum length of 65535 octets. The minimum number of octets shall be used.

- Value: The value is encoded in zero or more octets

### 9.3.2 Certificate Content

TCS\_278 All certificates shall have the structure shown in the certificate profile in Table 46.

<i>Field</i>	<i>Field ID</i>	<i>Tag</i>	<i>Length (bytes)</i>	<i>ASN.1 data type (see Appendix 1)</i>
ECC Certificate	C	'7F 21'	var	
ECC Certificate Body	B	'7F 4E'	var	
Certificate Profile Identifier	CPI	'5F 29'	'01'	INTEGER(0..255)
Certificate Authority Reference	CAR	'42'	'08'	KeyIdentifier
Certificate Holder Authorisation	CHA	'5F 4C'	'07'	CertificateHolderAuth orisation
Public Key	PK	'7F 49'	var	
Domain Parameters	DP	'06'	var	OBJECT IDENTIFIER
Public Point	PP	'86'	var	OCTET STRING
Certificate Holder Reference	CHR	'5F 20'	'08'	KeyIdentifier
Certificate Effective Date	CEfD	'5F 25'	'04'	TimeReal
Certificate Expiration Date	CExD	'5F 24'	'04'	TimeReal
ECC Certificate Signature	S	'5F 37'	var	OCTET STRING

**Table 46 Certificate Profile version 1**

Note: the Field ID will be used in later sections of this Appendix to indicate individual fields of a certificate, e.g. X.CAR is the Certificate Authority Reference mentioned in the certificate of user X.

#### 9.3.2.1 Certificate Profile Identifier

TCS\_279 Certificates shall use a Certificate Profile Identifier to indicate the certificate profile used. Version 1, as specified in Table 46, shall be identified by a value of '00'.

#### 9.3.2.2 Certificate Authority Reference

TCS\_280 The Certificate Authority Reference shall be used to identify the public key to be used to verify the certificate signature. The Certificate Authority Reference shall therefore be equal to the Certificate Holder Reference in the certificate of the corresponding certificate authority.

TCS\_281 An ERCA root certificate shall be self-signed, i.e., the Certificate Authority Reference and the Certificate Holder Reference in the certificate shall be equal.



TCS\_282 For an ERCA link certificate, the Certificate Holder Reference shall be equal to the CHR of the new ERCA root certificate. The Certificate Authority Reference for a link certificate shall be equal to the CHR of the previous ERCA root certificate.

#### 9.3.2.3 Certificate Holder Authorisation

TCS\_283 The Certificate Holder Authorisation shall be used to identify the type of certificate. It consists of the six most significant bytes of the Tachograph Application ID, concatenated with the type of equipment for which the certificate is intended.

#### 9.3.2.4 Public Key

The Public Key nests two data elements: the standardized domain parameters to be used with the public key in the certificate and the value of the public point.

TCS\_284 The data element Domain Parameters shall contain one of the object identifiers specified in Table 43 to reference a set of standardized domain parameters.

TCS\_285 The data element Public Point shall contain the public point. Elliptic curve public points shall be converted to octet strings as specified in [TR-03111]. The uncompressed encoding format shall be used. When recovering an elliptic curve point from its encoded format, the validations described in [TR-03111] shall always be carried out.

#### 9.3.2.5 Certificate Holder Reference

TCS\_286 The Certificate Holder Reference is an identifier for the public key provided in the certificate. It shall be used to reference this public key in other certificates.

TCS\_287 For card certificates and external GNSS facility certificates, the Certificate Holder Reference shall have the `ExtendedSerialNumber` data type specified in Appendix 1.

TCS\_288 For vehicle units, the manufacturer, when requesting a certificate, may or may not know the manufacturer-specific serial number of the VU for which that certificate and the associated private key is intended. In the first case, the Certificate Holder Reference shall have the `ExtendedSerialNumber` data type specified in Appendix 1. In the latter case, the Certificate Holder Reference shall have the `CertificateRequestID` data type specified in Appendix 1.

TCS\_289 For ERCA and MSCA certificates, the Certificate Holder Reference shall have the `CertificationAuthorityKID` data type specified in Appendix 1.

#### 9.3.2.6 Certificate Effective Date

TCS\_290 The Certificate Effective Date shall indicate the starting date and time of the validity period of the certificate. The Certificate Effective Date shall be the date of the certificate generation.

#### 9.3.2.7 Certificate Expiration Date

TCS\_291 The Certificate Expiration Date shall indicate the end date and time of the validity period of the certificate.

#### 9.3.2.8 Certificate Signature

TCS\_292 The signature on the certificate shall be created over the encoded certificate body, including the certificate body tag and length. The signature algorithm shall be ECDSA, as specified in [DSS], using the hashing algorithm linked to the key size of the signing authority, as specified in TCS\_192. The signature format shall be plain, as specified in [TR-03111].

#### 9.3.3 Requesting Certificates

TCS\_293 When requesting a certificate, a requester shall send the following data to its Certificate Authority:

- The Certificate Profile Identifier of the requested certificate
- The Certificate Authority Reference expected to be used for signing the certificate.
- The Public Key to be signed

TCS\_294 In addition to the data in TCS\_293, an MSCA shall send the following data in a certificate request to the ERCA, allowing the ERCA to create the Certificate Holder Reference of the new MSCA certificate:

- The numerical nation code of the Certification Authority (data type `NationNumeric` defined in Appendix 1)
- The alphanumerical nation code of the Certification Authority (data type `NationAlpha` defined in Appendix 1)
- The 1-byte serial number to distinguish the different keys of the Certification Authority in the case keys are changed
- The two-byte field containing Certification Authority specific additional info

TCS\_295 In addition to the data in TCS\_293, an equipment manufacturer shall send the following data in a certificate request to an MSCA, allowing the MSCA to create the Certificate Holder Reference of the new equipment certificate:

- A manufacturer-specific identifier of the equipment type
- If known (see TCS\_296), a serial number for the equipment, unique for the manufacturer, the equipment's type and the month of manufacturing. Otherwise, a unique certificate request identifier.
- The month and the year of equipment manufacturing or of the certificate request.

The manufacturer shall ensure that this data is correct and that the certificate returned by the MSCA is inserted in the intended equipment.

TCS\_296 In the case of a VU, the manufacturer, when requesting a certificate, may or may not know the manufacturer-specific serial number of the VU for which that certificate and the associated private key is intended. If known, the VU manufacturer shall send the serial number to the MSCA. If not known, the manufacturer shall uniquely identify each certificate request and send this certificate request serial number to the MSCA. The resulting certificate will then contain the certificate request serial number. After inserting the certificate in a specific VU, the manufacturer shall communicate the connection between the certificate request serial number and the VU identification to the MSCA.

## 10. VU- Card Mutual Authentication and Secure Messaging

### 10.1. General

TCS\_297 On a high level, secure communication between a vehicle unit and a tachograph card shall be based on the following steps:

- First, each party shall demonstrate to the other that it owns a valid public key certificate, signed by a Member State Certificate Authority. In turn, the MSCA public key certificate must be signed by the European root certificate authority. This step is called certificate chain verification and is specified in detail in section 10.2
- Second, the vehicle unit shall demonstrate to the card that it is in possession of the private key corresponding to the public key in the presented certificate. It does so by signing a random number sent by the card. The card verifies the signature over the random number. If this verification is successful, the VU is authenticated. This step is called VU Authentication and is specified in detail in section 10.3.
- Third, both parties independently calculate two AES session keys using an asymmetric key agreement algorithm. Using one of these session keys, the card creates a message authentication code (MAC) over some data sent by the VU. The VU verifies the MAC. If this verification is successful, the card is authenticated. This step is called Card Authentication and is specified in detail in section 10.4.
- Fourth, the VU and the card shall use the agreed session keys to ensure the confidentiality, integrity and authenticity of all exchanged messages. This is called Secure Messaging and is specified in detail in section 10.5.

TCS\_298 The mechanism described in TCS\_297 shall be triggered by the vehicle unit whenever a card is inserted into one of its card slots.

### 10.2. Mutual Certificate Chain Verification

#### 10.2.1 Card Certificate Chain Verification by VU

TCS\_299 Vehicle units shall use the protocol depicted in Figure 4 for verifying a tachograph card's certificate chain.

Notes to Figure 4:

- The Card certificates and public keys mentioned in the figure are those for mutual authentication. Section 9.1.5 denotes these as Card\_MA.
- The Card.CA certificates and public keys mentioned in the figure are those for signing card certificates and it is indicated in the CAR of the Card certificate. Section 9.1.3 denotes these as MSCA\_Card.
- The Card.CA.EUR certificate mentioned in the figure is the European root certificate that is indicated in the CAR of the Card.CA certificate.
- The Card.Link certificate mentioned in the figure is the card's link certificate, if present. As specified in section 9.1.2, this is a link certificate for a new European root key pair created by the ERCA and signed by the previous European private key.
- The Card.Link.EUR certificate is the European root certificate that is indicated in the CAR of the Card.Link certificate.

TCS\_300 As depicted in Figure 4, verification of the card's certificate chain shall begin upon card insertion. The vehicle unit shall read the card holder reference (`cardExtendedSerialNumber`) from EF ICC. The VU shall check if it knows the card, i.e., if it has successfully verified the card's certificate chain in the past and stored it for future reference. If it does, and the card certificate is still valid, the process continues with the verification of the VU certificate chain. Otherwise, the VU shall successively read from the card the MSCA\_Card certificate to be used for verifying the card certificate, the Card.CA. EUR certificate to be used for verifying the MSCA\_Card certificate, and possibly the link certificate, until it finds a certificate it knows or it can verify. If such a certificate is found, the VU shall use that certificate to verify the underlying card certificates it has read from the card. If successful, the process continues with the verification of the VU certificate chain. If not successful, the VU shall ignore the card.

Note: There are three ways in which the VU may know the Card.CA.EUR certificate:

- the Card.CA.EUR certificate is the same certificate as the VU's own EUR certificate;
- the Card.CA.EUR certificate precedes the VU's own EUR certificate and the VU contained this certificate already at issuance (see CSM\_81);
- the Card.CA.EUR certificate succeeds the VU's own EUR certificate and the VU received a link certificate in the past from another tachograph card, verified it and stored it for future reference.

TCS\_301 As indicated in Figure 4, once the VU has verified the authenticity and validity of a previously unknown certificate, it may store this certificate for future reference, such that it does not need to verify that certificate's authenticity again if it is presented to the VU again. Instead of storing the entire certificate, a VU may choose to store only the contents of the Certificate Body, as specified in section 9.3.2.

TCS\_302 The VU shall verify the temporal validity of any certificate read from the card or stored in its memory, and shall reject expired certificates. For verifying the temporal validity of a certificate presented by the card a VU shall use its internal clock.

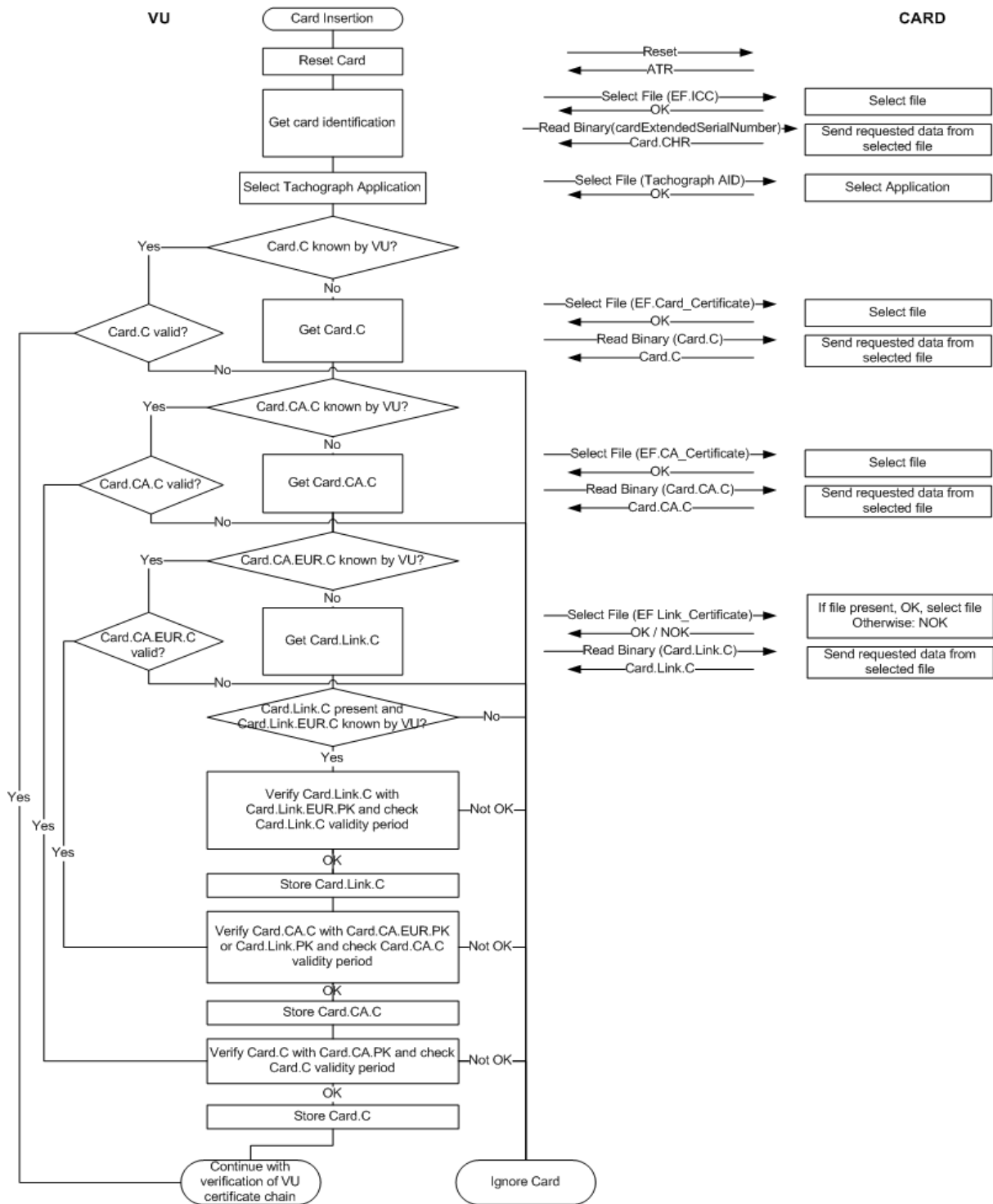
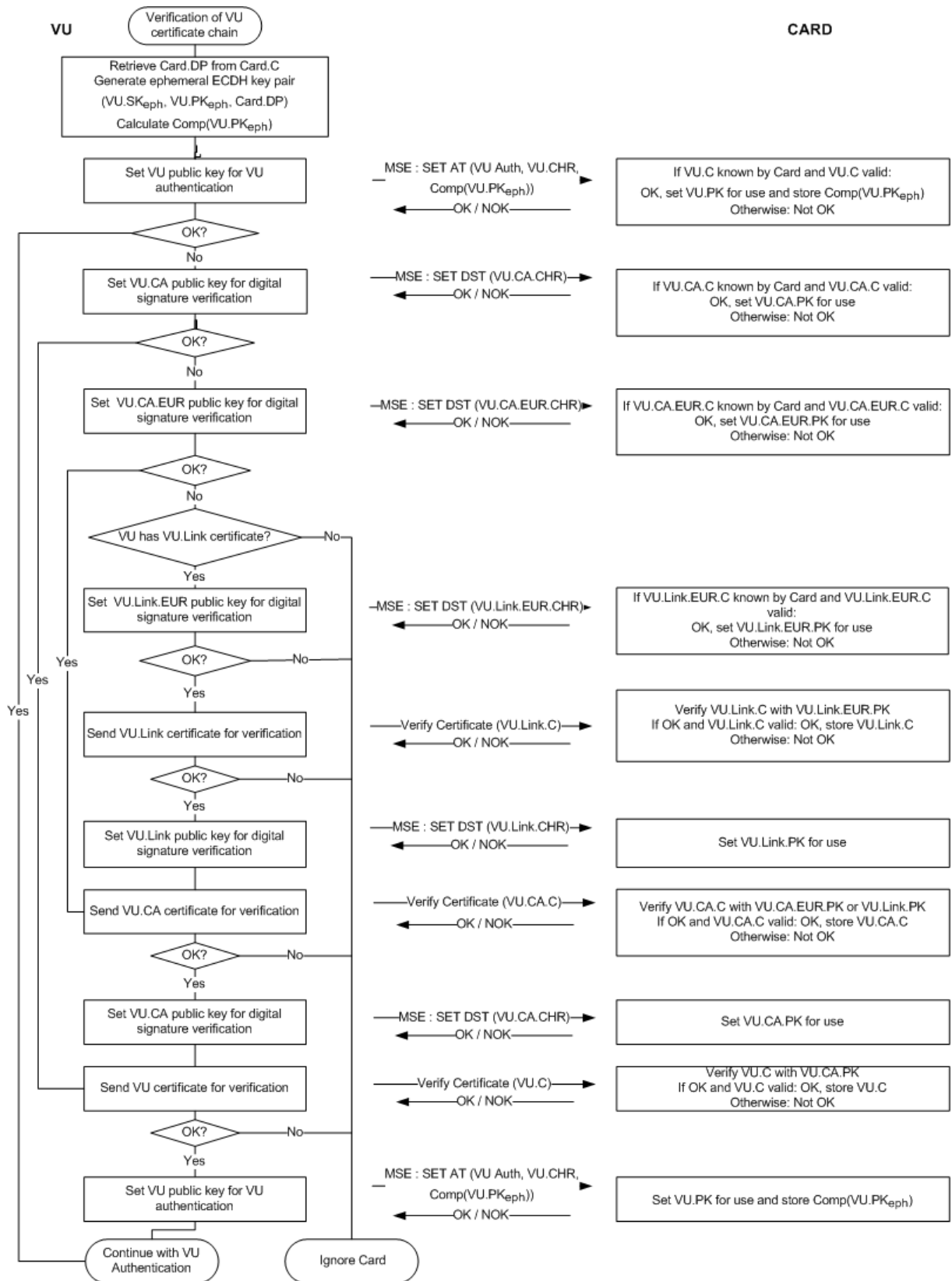


Figure 4 Protocol for Card Certificate Chain Verification by VU

10.2.2 VU Certificate Chain Verification by Card

TCS\_303 Tachograph cards shall use the protocol depicted in Figure 5 for verifying a VU's certificate chain.



**Figure 5 Protocol for VU Certificate Chain Verification by Card**

Notes to Figure 5:

- The VU certificates and public keys mentioned in the figure are those for mutual authentication. Section 9.1.4 denotes these as VU\_MA.
- The VU.CA certificates and public keys mentioned in the figure are those for signing VU and external GNSS facility certificates. Section 9.1.3 denotes these as MSCA\_VU-EGF.
- The VU.CA.EUR certificate mentioned in the figure is the European root certificate that is indicated in the CAR of the VU.CA certificate.
- The VU.Link certificate mentioned in the figure is the VU's link certificate, if present. As specified in section 9.1.2, this is a link certificate for a new European root key pair created by the ERCA and signed by the previous European private key.
- The VU.Link.EUR certificate is the European root certificate that is indicated in the CAR of the VU.Link certificate.

TCS\_304

As depicted in Figure 5, verification of the certificate chain of the vehicle unit shall begin with the vehicle unit attempting to set its own public key for use in the tachograph card. If this succeeds, it means that the card successfully verified the VU's certificate chain in the past, and has stored the VU certificate for future reference. In this case, the VU certificate is set for use and the process continues with VU Authentication. If the card does not know the VU certificate, the VU shall successively present the VU.CA certificate to be used for verifying its VU certificate, the VU.CA.EUR certificate to be used for verifying the VU.CA certificate, and possibly the link certificate, in order to find a certificate known or verifiable by the card. If such a certificate is found, the card shall use that certificate to verify the underlying VU certificates presented to it. If successful, the VU shall finally set its public key for use in the tachograph card. If not successful, the VU shall ignore the card.

Note: There are three ways in which the card may know the VU.CA.EUR certificate:

- the VU.CA.EUR certificate is the same certificate as the card's own EUR certificate;
- the VU.CA.EUR certificate precedes the card's own EUR certificate and the card contained this certificate already at issuance (see CSM\_91);
- the VU.CA.EUR certificate succeeds the card's own EUR certificate and the card received a link certificate in the past from another vehicle unit, verified it and stored it for future reference.

TCS\_305

The VU shall use the MSE: Set AT command to set its public key for use in the tachograph card. As specified in Appendix 2, this command contains an indication of the cryptographic mechanism that will be used with the key that is set. This mechanism shall be 'VU Authentication using the ECDSA algorithm, in combination with the hashing algorithm linked to the key size of the VU's VU\_MA key pair, as specified in TCS\_192'.



TCS\_306 The MSE: Set AT command also contains an indication of the ephemeral key pair which the VU will use during session key agreement (see section 10.4). Therefore, before sending the MSE: Set AT command, the VU shall generate an ephemeral ECC key pair. For generating the ephemeral key pair, the VU shall use the standardized domain parameters indicated in the card certificate. The ephemeral key pair is denoted as  $(VU.SK_{eph}, VU.PK_{eph}, Card.DP)$ . The VU shall take the x-coordinate of the ECDH ephemeral public point as the key identification; this is called the compressed representation of the public key and denoted as  $Comp(VU.PK_{eph})$ .

TCS\_307 If the MSE: Set AT command is successful, the card shall set the indicated  $VU.PK$  for subsequent use during Vehicle Authentication, and shall temporarily store  $Comp(VU.PK_{eph})$ . In case two or more successful MSE: Set AT commands are sent before session key agreement is performed, the card shall store only the last  $Comp(VU.PK_{eph})$  received.

TCS\_308 The card shall verify the temporal validity of any certificate presented by the VU or referenced by the VU while stored in the card's memory, and shall reject expired certificates.

TCS\_309 For verifying the temporal validity of a certificate presented by the VU, each tachograph card shall internally store some data representing the current time. This data shall not be directly updatable by a VU. At issuance, the current time of a card shall be set equal to the Effective Date of the card's  $Card\_MA$  certificate. A card shall update its current time if the Effective Date of an authentic 'valid source of time' certificate presented by a VU is more recent than the card's current time. In that case, the card shall set its current time to the Effective Date of that certificate. The card shall accept only the following certificates as a valid source of time:

- Second-generation ERCA link certificates
- Second-generation MSCA certificates
- Second-generation VU certificates issued by the same country as the card's own card certificate(s).

Note: the last requirement implies that a card shall be able to recognize the CAR of the VU certificate, i.e. the  $MSCA\_VU$ -EGF certificate. This will not be the same as the CAR of its own certificate, which is the  $MSCA\_Card$  certificate.

TCS\_310 As indicated in Figure 5, once the card has verified the authenticity and validity of a previously unknown certificate, it may store this certificate for future reference, such that it does not need to verify that certificate's authenticity again if it is presented to the card again. Instead of storing the entire certificate, a card may choose to store only the contents of the Certificate Body, as specified in section 9.3.2.

### 10.3. VU Authentication

TCS\_311 Vehicle units and cards shall use the VU Authentication protocol depicted in Figure 6 to authenticate the VU towards the card. VU Authentication enables the tachograph card to explicitly verify that the VU is authentic. To do so, the VU shall use its private key to sign a challenge generated by the card.

TCS\_312 Next to the card challenge, the VU shall include in the signature the card holder reference taken from the card certificate.

Note: This ensures that the card to which the VU authenticates itself is the same card whose certificate chain the VU has verified previously.

TCS\_313

The VU shall also include in the signature the identifier of the ephemeral public key  $\text{Comp}(\text{VU.PK}_{\text{eph}})$  which the VU will use to set up Secure Messaging during the Chip Authentication process specified in section 10.4.

Note: This ensures that the VU with which a card communicates during a Secure Messaging session is the same VU that was authenticated by the card.

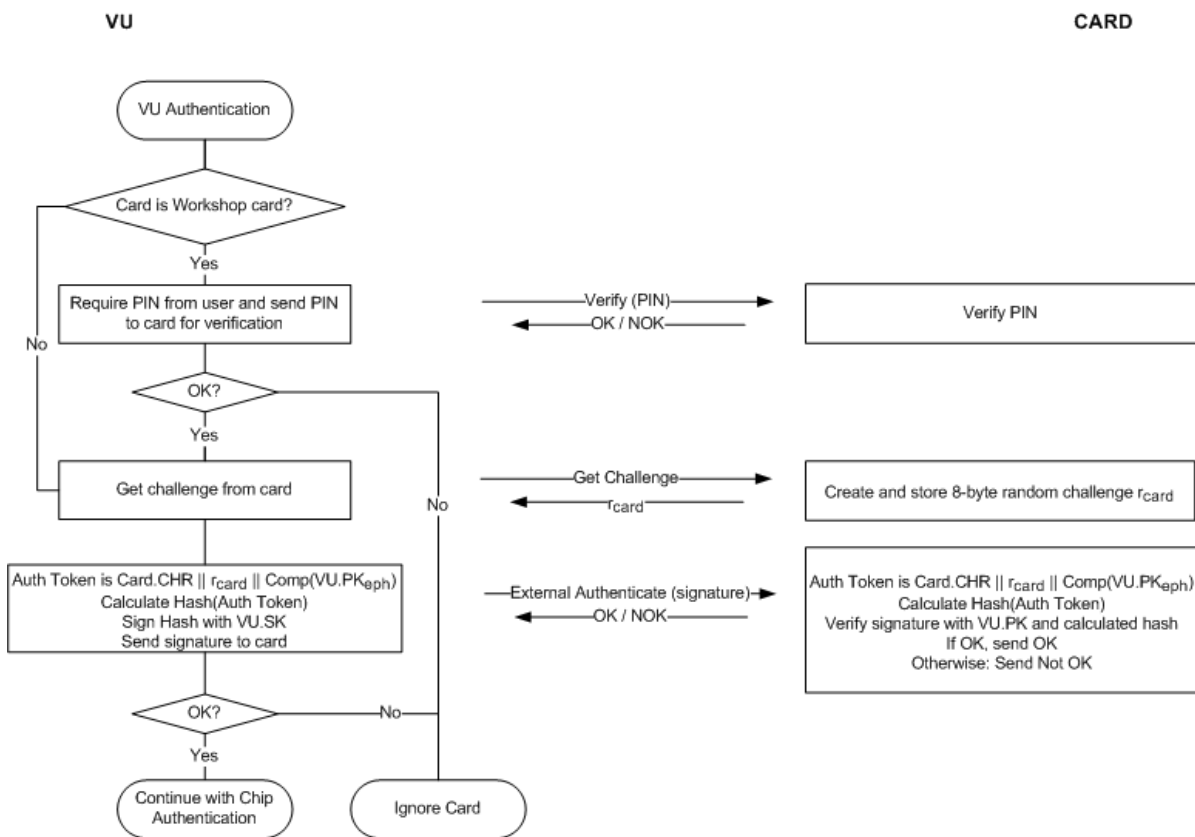


Figure 6 VU Authentication protocol

TCS\_314

If multiple GET CHALLENGE commands are sent by the VU during VU Authentication, the card shall return a new 8-byte random challenge each time, but shall store only the last challenge.

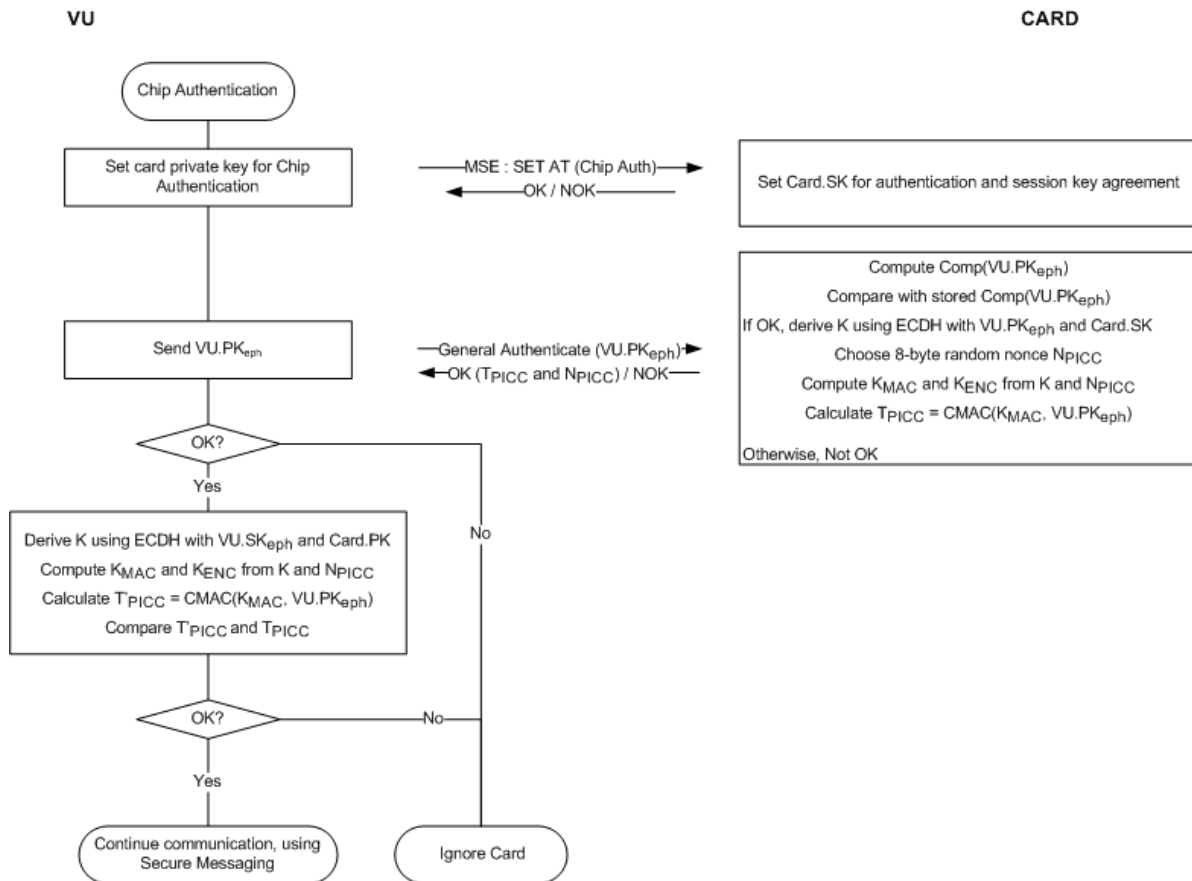
TCS\_315

The signing algorithm used by the VU for VU Authentication shall be ECDSA as specified in [DSS], using the hashing algorithm linked to the key size of the VU's VU\_MA key pair, as specified in TCS\_192. The signature format shall be plain, as specified in [TR-03111]. The VU shall send the resulting signature to the card.

- TCS\_316 Upon receiving the VU’s signature in an EXTERNAL AUTHENTICATE command, the card shall
- Calculate the authentication token by concatenating Card.CHR, the card challenge  $r_{card}$  and the identifier of the VU ephemeral public key  $Comp(VU.PK_{eph})$ ,
  - Calculate the hash over the authentication token, using the hashing algorithm linked to the key size of the VU’s VU\_MA key pair, as specified in TCS\_192,
  - Verify the VU’s signature using the ECDSA algorithm in combination with VU.PK and the calculated hash.

10.4. Chip Authentication and Session Key Agreement

TCS\_317 Vehicle units and cards shall use the Chip Authentication protocol depicted in **Figure 7** to authenticate the card towards the VU. Chip Authentication enables the vehicle unit to explicitly verify that the card is authentic.



**Figure 7 Chip Authentication and session key agreement**

TCS\_318 The VU and the card shall take the following steps:

1. The vehicle unit initiates the Chip Authentication process by sending the MSE: Set AT command indicating ‘Chip Authentication using the ECDH algorithm resulting in an AES session key length linked to the key size of the card’s Card\_MA key pair, as specified in TCS\_192’. The VU shall determine the key size of the card’s key pair from the card certificate.

2. The VU sends the public point  $VU.PK_{eph}$  of its ephemeral key pair to the card. As explained in TCS\_306, the VU generated this ephemeral key pair prior to the verification of the VU certificate chain. The VU sent the identifier of the ephemeral public key  $Comp(VU.PK_{eph})$  to the card, and the card stored it.
3. The card computes  $Comp(VU.PK_{eph})$  from  $VU.PK_{eph}$  and compares this to the stored value of  $Comp(VU.PK_{eph})$ .
4. Using the ECDH algorithm in combination with the card's static private key and the VU's ephemeral public key, the card computes a secret  $K$ .
5. The card chooses a random 8-byte nonce  $N_{PICC}$  and uses it to derive two AES session keys  $K_{MAC}$  and  $K_{ENC}$  from  $K$ . See TCS\_321.
6. Using  $K_{MAC}$ , the card computes an authentication token over the VU ephemeral public key identifier:  $T_{PICC} = CMAC(K_{MAC}, VU.PK_{eph})$ . The card sends  $N_{PICC}$  and  $T_{PICC}$  to the vehicle unit.
7. Using the ECDH algorithm in combination with the card's static public key and the VU's ephemeral private key, the VU computes the same secret  $K$  as the card did in step 4.
8. The VU derives session keys  $K_{MAC}$  and  $K_{ENC}$  from  $K$  and  $N_{PICC}$ ; see TCS\_321.
9. The VU verifies the authentication token  $T_{PICC}$ .

TCS\_319 In step 3 above, the card shall compute  $Comp(VU.PK_{eph})$  as the x-coordinate of the public point in  $VU.PK_{eph}$ .

TCS\_320 In steps 4 and 7 above, the card and the vehicle unit shall use the ECKA-EG algorithm as defined in [TR-03111].

TCS\_321 In steps 5 and 8 above, the card and the vehicle unit shall use the key derivation function for AES session keys defined in [TR-03111], with the following precisions and changes:

- The value of the counter shall be '00 00 00 01' for  $K_{ENC}$  and '00 00 00 02' for  $K_{MAC}$ .
- The optional nonce  $r$  shall be used and shall be equal to  $N_{PICC}$ .
- For deriving 128-bits AES keys, the hashing algorithm to be used shall be SHA-256.
- For deriving 192-bits AES keys, the hashing algorithm to be used shall be SHA-384.
- For deriving 256-bits AES keys, the hashing algorithm to be used shall be SHA-512.

The length of the session keys (i.e. the length at which the hash is truncated) shall be linked to the size of the Card\_MA key pair, as specified in TCS\_192.

TCS\_322 In steps 6 and 9 above, the card and the vehicle unit shall use the AES algorithm in CMAC mode, as specified in [SP 800-38B]. The length of  $T_{PICC}$  shall be linked to the length of the AES session keys, as specified in TCS\_192.

## 10.5. Secure Messaging

### 10.5.1 General

TCS\_323 All commands and responses exchanged between a vehicle unit and a tachograph card after successful Chip Authentication took place and until the end of the session shall be protected by Secure Messaging.

- TCS\_324 Except when reading from a file with access condition SM-R-ENC-MAC-G2 (see Appendix 2, section 4), Secure Messaging shall be used in authentication-only mode. In this mode, a cryptographic checksum (a.k.a. MAC) is added to all commands and responses to ensure message authenticity and integrity.
- TCS\_325 When reading data from a file with access condition SM-R-ENC-MAC-G2, Secure Messaging shall be used in encrypt-then-authenticate mode, i.e. the response data is encrypted first to ensure message confidentiality, and afterwards a MAC over the formatted encrypted data is calculated to ensure authenticity and integrity.
- TCS\_326 Secure Messaging shall use AES as defined in [AES] with the session keys  $K_{MAC}$  and  $K_{ENC}$  that were agreed during Chip Authentication.
- TCS\_327 An unsigned integer shall be used as the Send Sequence Counter (SSC) to prevent replay attacks. The size of the SSC shall be equal to the AES block size, i.e. 128 bits. The SSC shall be in MSB-first format. The Send Sequence Counter shall be initialized to zero (i.e. '00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00') when Secure Messaging is started. The SSC shall be increased every time before a command or response APDU is generated, i.e. since the starting value of the SSC in a SM session is 0, in the first command the value of the SSC will be 1. The value of SSC for the first response will be 2.
- TCS\_328 For message encryption,  $K_{ENC}$  shall be used with AES in the Cipher Block Chaining (CBC) mode of operation, as defined in [ISO 10116], with an interleave parameter  $m = 1$  and an initialization vector  $SV = E(K_{ENC}, SSC)$ , i.e. the current value of the Send Sequence Counter encrypted with  $K_{ENC}$ .
- TCS\_329 For message authentication,  $K_{MAC}$  shall be used with AES in CMAC mode as specified in [SP 800-38B]. The length of the MAC shall be linked to the length of the AES session keys, as specified in TCS\_192. The Send Sequence Counter shall be included in the MAC by prepending it before the datagram to be authenticated.
- 10.5.2 Secure Message Structure
- TCS\_330 Secure Messaging shall make use only of the Secure Messaging data objects (see [ISO 7816-4]) listed in Table 47. In any message, these data objects shall be used in the order specified in this table.

<i>Data Object Name</i>	<i>Tag</i>	<i>Presence (M)andatory, (C)onditional or (F)orbidden in</i>	
		<b>Commands</b>	<b>Responses</b>
Plain value not encoded in BER-TLV	'81'	C	C
Plain value encoded in BER-TLV, but not including SM DOs	'B3'	C	C
Padding-content indicator followed by cryptogram, plain value not encoded in BER-TLV	'87'	C	C
Protected Le	'97'	C	F
Processing Status	'99'	F	M

<i>Data Object Name</i>	<i>Tag</i>	<i>Presence (M)andatory, (C)onditional or (F)orbidden in</i>	
		<b>Commands</b>	<b>Responses</b>
Cryptographic Checksum	'8E'	M	M

**Table 47 Secure Messaging Data Objects**

Note: As specified in Appendix 2, tachograph cards may support the READ BINARY and UPDATE BINARY command with an odd INS byte ('B1' resp. 'D7'). These command variants are required to read and update files with more than 32768 bytes or more. In case such a variant is used, a data object with tag 'B3' shall be used instead of an object with tag '81'. See Appendix 2 for more information.

- TCS\_331 All SM data objects shall be encoded in DER TLV as specified in [ISO 8825-1]. This encoding results in a Tag-Length-Value (TLV) structure as follows:
- Tag: The tag is encoded in one or two octets and indicates the content.
  - Length: The length is encoded as an unsigned integer in one, two, or three octets, resulting in a maximum length of 65535 octets. The minimum number of octets shall be used.
  - Value: The value is encoded in zero or more octets
- TCS\_332 APDUs protected by Secure Messaging shall be created as follows:
- The command header shall be included in the MAC calculation, therefore value '0C' shall be used for the class byte CLA.
  - As specified in Appendix 2, all INS bytes shall be even, with the possible exception of odd INS bytes for the READ BINARY and UPDATE BINARY commands.
  - The actual value of Lc will be modified to Lc' after application of secure messaging.
  - The Data field shall consist of SM data objects.
  - In the protected command APDU the new Le byte shall be set to '00'. If required, a data object '97' shall be included in the Data field in order to convey the original value of Le.
- TCS\_333 Any data object to be encrypted shall be padded according to [ISO 7816-4] using padding-content indicator '01'. For the calculation of the MAC, each data object in the APDU shall also be separately padded according to [ISO 7816-4].

Note: Padding for Secure Messaging is always performed by the secure messaging layer, not by the CMAC or CBC algorithms.

### **Summary and Examples**

A command APDU with applied Secure Messaging will have the following structure, depending on the case of the respective unsecured command (DO is data object):

- Case 1: CLA INS P1 P2 || Lc' || DO '8E' || Le
- Case 2: CLA INS P1 P2 || Lc' || DO '97' || DO'8E' || Le
- Case 3 (even INS byte): CLA INS P1 P2 || Lc' || DO '81' || DO'8E' || Le
- Case 3 (odd INS byte): CLA INS P1 P2 || Lc' || DO 'B3' || DO'8E' || Le
- Case 4 (even INS byte): CLA INS P1 P2 || Lc' || DO '81' || DO'97' || DO'8E' || Le
- Case 4 (odd INS byte): CLA INS P1 P2 || Lc' || DO 'B3' || DO'97' || DO'8E' || Le

where Le = '00' or '00 00' depending on whether short length fields or extended length fields are used; see [ISO 7816-4].

A response APDU with applied Secure Messaging will have the following structure, depending on the case of the respective unsecured response:

- Case 1 or 3: DO '99' || DO '8E' || SW1SW2
- Case 2 or 4 (even INS byte) with encryption: DO '81' || DO '99' || DO '8E' || SW1SW2
- Case 2 or 4 (even INS byte) without encryption: DO '87' || DO '99' || DO '8E' || SW1SW2
- Case 2 or 4 (odd INS byte) without encryption: DO 'B3' || DO '99' || DO '8E' || SW1SW2

Note: Case 2 or 4 (odd INS byte) with encryption is never used in the communication between a VU and a card.

Below are three example APDU transformations for commands with even INS code. Figure 8 shows an authenticated Case 4 command APDU, Figure 9 shows an authenticated Case 2/Case 4 response APDU, and Figure 10 shows an encrypted and authenticated Case 2/Case 4 response APDU.

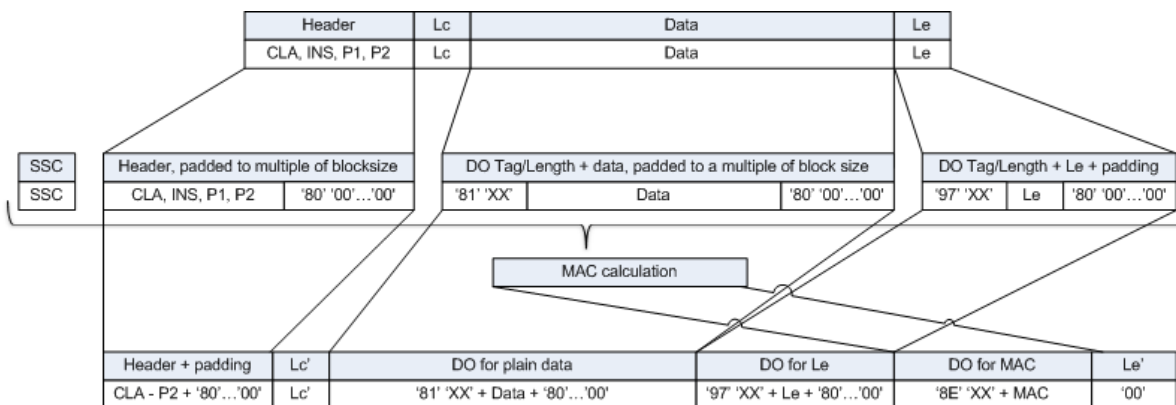


Figure 8 Transformation of an authenticated Case 4 Command APDU

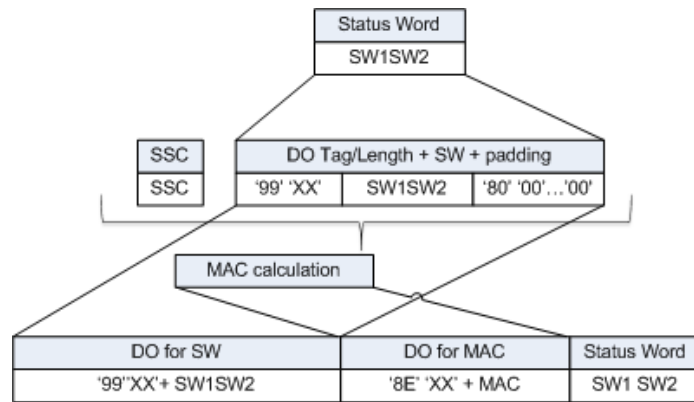


Figure 9 Transformation of an authenticated Case 1 / Case 3 Response APDU

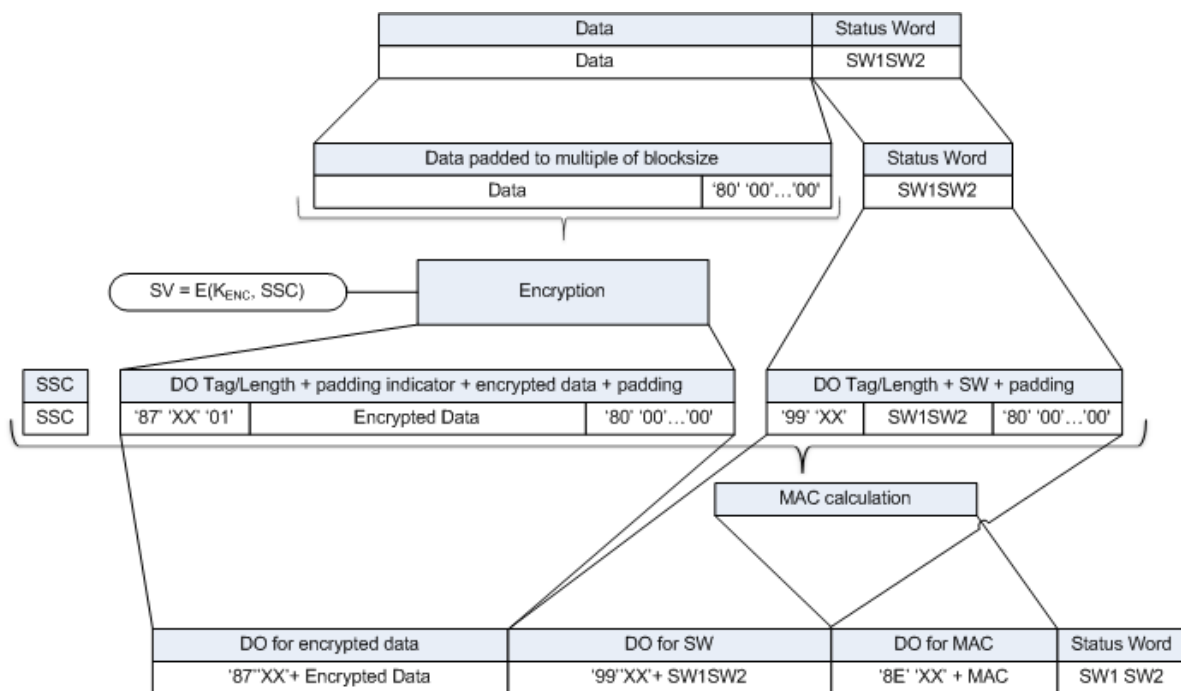


Figure 10 Transformation of an encrypted and authenticated Case 2/Case 4 Response APDU

### 10.5.3 Secure Messaging Session Abortion

TCS\_334 A vehicle unit shall abort an ongoing Secure Messaging session if and only if one of the following conditions occur:



- it receives a plain response APDU,
- it detects a Secure Messaging error in a response APDU:
  - o An expected Secure Messaging data object is missing, the order of data objects is incorrect, or an unknown data object is included.
  - o A Secure Messaging data object is incorrect, e.g. the MAC value is incorrect, the TLV structure is incorrect or the padding indicator in tag '87' is not equal to '01'.
- the card sends a status byte indicating it detected an SM error (see TCS\_336),
- the limit for the number of commands and associated responses within the current session is reached. For a given VU, this limit shall be defined by its manufacturer, taking into account the security requirements of the hardware used, with a maximum value of 240 SM commands and associated responses per session.

TCS\_335 A tachograph card shall abort an ongoing Secure Messaging session if and only if one of the following conditions occur:

- it receives a plain command APDU,
- it detects a Secure Messaging error in a command APDU:
  - o An expected Secure Messaging data object is missing, the order of data objects is incorrect, or an unknown data object is included.
  - o A Secure Messaging data object is incorrect, e.g. the MAC value is incorrect or the TLV structure is incorrect.
- it is depowered or reset,
- the VU selects an application on the card,
- the VU starts the VU Authentication process,
- the limit for the number of commands and associated responses within the current session is reached. For a given card, this limit shall be defined by its manufacturer, taking into account the security requirements of the hardware used, with a maximum value of 240 SM commands and associated responses per session.

TCS\_336 Regarding SM error handling by a tachograph card:

- If in a command APDU some expected Secure Messaging data objects are missing, the order of data objects is incorrect or unknown data objects are included, a tachograph card shall respond with status bytes '69 87'.
- If a Secure Messaging data object in a command APDU is incorrect, a tachograph card shall respond with status bytes '69 88'.

In such a case, the status bytes shall be returned without using SM.

TCS\_337 If a Secure Messaging session between a VU and a tachograph card is aborted, the VU and the tachograph card shall

- securely destroy the stored session keys
- immediately establish a new Secure Messaging session, as described in sections 10.2 - 10.5.

TCS\_338 If for any reason the VU decides to restart mutual authentication towards an inserted card, the process shall restart with verification of the card certificate chain, as described in section 10.2, and shall continue as described in sections 10.2 - 10.5.

## 11. VU – External GNSS Facility Coupling, Mutual Authentication and Secure Messaging

### 11.1. General

- TCS\_339 The GNSS facility used by a VU to determine its position may be internal, (i.e. built into the VU casing and not detachable), or it may be an external module. In the first case, there is no need to standardize the internal communication between the GNSS facility and the VU, and the requirements in this chapter do not apply. In the latter case, communication between the VU and the external GNSS facility shall be standardized and protected as described in this chapter.
- TCS\_340 Secure communication between a vehicle unit and an external GNSS facility shall take place in the same way as secure communication between a vehicle unit and a tachograph card, with the external GNSS facility (EGF) taking the role of the card. All requirements mentioned in chapter 10 for tachograph cards shall be satisfied by an EGF, taking into account the deviations, clarifications and additions mentioned in this chapter. In particular, mutual certificate chain verification, VU Authentication and Chip Authentication shall be performed as described in sections 11.3 and 11.4.
- TCS\_341 Communication between a vehicle unit and an EGF differs from communication between a vehicle unit and a card in the fact that a vehicle unit and an EGF must be coupled once in a workshop before the VU and the EGF can exchange GNSS-based data during normal operation. The coupling process is described in section 11.2.
- TCS\_342 For communication between a vehicle unit and an EGF, APDU commands and responses based on [ISO 7816-4] and [ISO 7816-8] shall be used. The exact structure of these APDUs is defined in Appendix 2 of this Annex.
- ### 11.2. VU and External GNSS Facility Coupling
- TCS\_343 A vehicle unit and an EGF in a vehicle shall be coupled by a workshop. Only a coupled vehicle unit and EGF shall be able to communicate during normal operation.
- TCS\_344 Coupling of a vehicle unit and an EGF shall only be possible if the vehicle unit is in calibration mode. The coupling shall be initiated by the vehicle unit.
- TCS\_345 A workshop may re-couple a vehicle unit to another EGF or to the same EGF at any time. During re-coupling, the VU shall securely destroy the existing EGF\_MA certificate in its memory and shall store the EGF\_MA certificate of the EGF to which it is being coupled.
- TCS\_346 A workshop may re-couple an external GNSS facility to another VU or to the same VU at any time. During re-coupling, the EGF shall securely destroy the existing VU\_MA certificate in its memory and shall store the VU\_MA certificate of the VU to which it is being coupled.

### 11.3. Mutual Certificate Chain Verification

#### 11.3.1 General

TCS\_347 Mutual certificate chain verification between a VU and an EGF shall take place only during the coupling of the VU and the EGF by a workshop. During normal operation of a coupled VU and EGF, no certificates shall be verified. Instead, the VU and EGF shall trust the certificates they stored during the coupling, after checking the temporal validity of these certificates. The VU and the EGF shall not trust any other certificates for protecting the VU – EGF communication during normal operation.

#### 11.3.2 During VU – EGF Coupling

TCS\_348 During the coupling to an EGF, a vehicle unit shall use the protocol depicted in Figure 4 (section 10.2.1) for verifying the external GNSS facility's certificate chain.

Notes to Figure 4 within this context:

- Communication control is out of the scope of this Appendix. However, an EGF is not a smart card and hence the VU will probably not send a Reset to initiate the communication and will not receive an ATR.
- The Card certificates and public keys mentioned in the figure shall be interpreted as the EGF's certificates and public keys for mutual authentication. Section 9.1.6 denotes these as EGF\_MA.
- The Card.CA certificates and public keys mentioned in the figure shall be interpreted as the MSCA's certificates and public keys for signing EGF certificates. Section 9.1.3 denotes these as MSCA\_VU-EGF.
- The Card.CA.EUR certificate mentioned in the figure shall be interpreted as the European root certificate that is indicated in the CAR of the MSCA\_VU-EGF certificate.
- The Card.Link certificate mentioned in the figure shall be interpreted as the EGF's link certificate, if present. As specified in section 9.1.2, this is a link certificate for a new European root key pair created by the ERCA and signed by the previous European private key.
- The Card.Link.EUR certificate is the European root certificate that is indicated in the CAR of the Card.Link certificate.
- Instead of the cardExtendedSerialNumber, the VU shall read the sensorGNSSserialNumber from EF ICC.
- Instead of selecting the Tachograph AID, the VU shall select the EGF AID.
- 'Ignore Card' shall be interpreted as 'Ignore EGF'.

TCS\_349 Once it has verified the EGF\_MA certificate, the vehicle unit shall store this certificate for use during normal operation; see section 11.3.3.

TCS\_350 During the coupling to a VU, an external GNSS unit shall use the protocol depicted in Figure 5 (section 10.2.2) for verifying the VU's certificate chain.

Notes to Figure 5 within this context:

- The VU shall generate a fresh ephemeral key pair using the domain parameters in the EGF certificate.
- The VU certificates and public keys mentioned in the figure are those for mutual authentication. Section 9.1.4 denotes these as VU\_MA.

- The VU.CA certificates and public keys mentioned in the figure are those for signing VU and external GNSS facility certificates. Section 9.1.3 denotes these as MSCA\_VU-EGF.
- The VU.CA.EUR certificate mentioned in the figure is the European root certificate that is indicated in the CAR of the VU.CA certificate.
- The VU.Link certificate mentioned in the figure is the VU's link certificate, if present. As specified in section 9.1.2, this is a link certificate for a new European root key pair created by the ERCA and signed by the previous European private key.
- The VU.Link.EUR certificate is the European root certificate that is indicated in the CAR of the VU.Link certificate.

TCS\_351 In deviation from requirement TCS\_309, an EGF shall use the GNSS time to verify the temporal validity of any certificate presented.

TCS\_352 Once it has verified the VU\_MA certificate, the external GNSS unit shall store this certificate for use during normal operation; see section 11.3.3.

#### 11.3.3 During Normal Operation

TCS\_353 During normal operation, a vehicle unit and an EGF shall use the protocol depicted in Figure 11 for verifying the temporal validity of the stored EGF\_MA and VU\_MA certificates and for setting the VU\_MA public key for subsequent VU Authentication. No further mutual verification of the certificate chains shall take place during normal operation.

Note that Figure 11 in essence consists of the first steps shown in Figure 4 and Figure 5. Again, note that since an EGF is not a smart card, the VU will probably not send a Reset to initiate the communication and will not receive an ATR. In any case this is out of the scope of this Appendix.

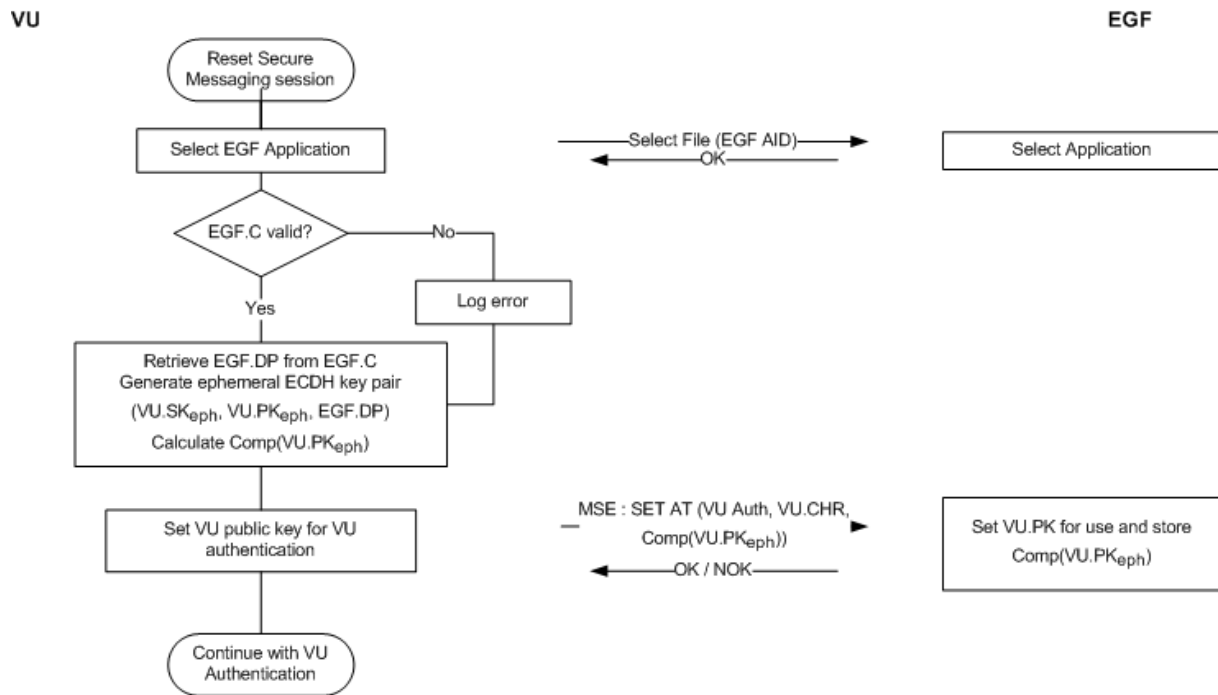


Figure 11 Mutual verification of certificate temporal validity during normal VU - EGF operation

TCS\_354 As shown in Figure 11, the vehicle unit shall log an error if the EGF\_MA certificate is no longer valid. However, mutual authentication, key agreement and subsequent communication via secure messaging shall proceed normally.

11.4. VU Authentication, Chip Authentication and Session Key Agreement

TCS\_355 VU Authentication, Chip Authentication and session key agreement between a VU and an EGF shall take place during coupling and whenever a Secure Messaging session is re-established during normal operation. The VU and the EGF shall carry out the processes described in sections 10.3 and 10.4. All requirements in these sections shall apply.

11.5. Secure Messaging

TCS\_356 All commands and responses exchanged between a vehicle unit and an external GNSS facility after successful Chip Authentication took place and until the end of the session shall be protected by Secure Messaging.in authentication-only mode. All requirements in section 10.5 shall apply.

TCS\_357 If a Secure Messaging session between a VU and an EGF is aborted, the VU shall immediately establish a new Secure Messaging session, as described in section 11.3.3 and 11.4.

## 12. VU – Motion Sensor Pairing and Communication

### 12.1. General

TCS\_358

A vehicle unit and a motion sensor shall communicate using the interface protocol specified in [ISO 16844-3] during pairing and in normal operation, with the changes described in this chapter and in section 9.2.1.

Note: readers of this chapter are supposed to be familiar with the contents of [ISO 16844-3].

### 12.2. VU – Motion Sensor Pairing Using Different Key Generations

As explained in section 9.2.1, the motion sensor master key and all associated keys are regularly replaced. This leads to the presence of up to three motion sensor-related AES keys  $K_{M-WC}$  (of consecutive key generations) in workshop cards. Similarly, in motion sensors up to three different AES-based encryptions of data (based on consecutive generations of the motion sensor master key  $K_M$ ) may be present. A vehicle unit contains only one motion sensor-related key  $K_{M-VU}$ .

TCS\_359

A second-generation VU and a second-generation motion sensor shall be paired as follows (compare Table 6 in [ISO 16844-3]):

1. A second-generation workshop card is inserted into the VU and the VU is connected to the motion sensor.
2. The VU reads all available  $K_{M-WC}$  keys from the workshop card, inspects their key version numbers and chooses the one matching the version number of the VU's  $K_{M-VU}$  key. If the matching  $K_{M-WC}$  key is not present on the workshop card, the VU aborts the pairing process and shows an appropriate error message to the workshop card holder.
3. The VU calculates the motion sensor master key  $K_M$  from  $K_{M-VU}$  and  $K_{M-WC}$ , and the identification key  $K_{ID}$  from  $K_M$ , as specified in section 9.2.1.
4. The VU sends the instruction to initiate the pairing process towards the motion sensor, as described in [ISO 16844-3], and encrypts the serial number it receives from the motion sensor with the identification key  $K_{ID}$ . The VU sends the encrypted serial number back to the motion sensor.
5. The motion sensor matches the encrypted serial number consecutively with each of the encryptions of the serial number it holds internally. If it finds a match, the VU is authenticated. The motion sensor notes the generation of  $K_{ID}$  used by the VU and returns the matching encrypted version of its pairing key; i.e. the encryption that was created using the same generation of  $K_M$ .
6. The VU decrypts the pairing key using  $K_M$ , generates a session key  $K_S$ , encrypts it with the pairing key and sends the result to the motion sensor. The motion sensor decrypts  $K_S$ .
7. The VU assembles the pairing information as defined in [ISO 16844-3], encrypts the information with the pairing key, and sends the result to the motion sensor. The motion sensor decrypts the pairing information.
8. The motion sensor encrypts the received pairing information with the received  $K_S$  and returns this to the VU. The VU verifies that the pairing information is the same information which the VU sent to the motion sensor in the previous step. If it is, this proves that the motion sensor used the same  $K_S$  as the VU and hence in step 5 sent its pairing key encrypted with the correct generation of  $K_M$ . Hence, the motion sensor is authenticated.

Note that steps 2 and 5 are different from the standard process in [ISO 16844-3]; the other steps are standard.

Example: Suppose a pairing takes place in the first year of the validity of the ERCA (3) certificate; see Figure 2 in section 9.2.2.2. Moreover

- Suppose the motion sensor was issued in the last year of the validity of the ERCA (1) certificate. It will therefore contain the following keys and data:
  - $N_s[1]$ : its serial number encrypted with generation 1 of  $K_{ID}$ ,
  - $N_s[2]$ : its serial number encrypted with generation 2 of  $K_{ID}$ ,
  - $N_s[3]$ : its serial number encrypted with generation 3 of  $K_{ID}$ ,
  - $K_P[1]$ : its generation-1 pairing key<sup>12</sup>, encrypted with generation 1 of  $K_M$ ,
  - $K_P[2]$ : its generation-2 pairing key, encrypted with generation 2 of  $K_M$ ,
  - $K_P[3]$ : its generation-3 pairing key, encrypted with generation 3 of  $K_M$ ,
- Suppose that the workshop card was issued in the first year of the validity of the ERCA (3) certificate. It will therefore contain the generation 2 and generation 3 of the  $K_{M-WC}$  key.
- Suppose the VU is a generation-2 VU, containing the generation 2 of  $K_{M-VU}$ .

In this case, the following will happen in steps 2 - 5:

- Step 2: The VU reads generation 2 and generation 3 of  $K_{M-WC}$  from the workshop card and inspects their version numbers.
- Step 3: The VU combines the generation-2  $K_{M-WC}$  with its  $K_{M-VU}$  to compute  $K_M$  and  $K_{ID}$ .
- Step 4: The VU encrypts the serial number it receives from the motion sensor with  $K_{ID}$ .
- Step 5: The motion sensor compares the received data with  $N_s[1]$  and doesn't find a match. Next, it compares the data with  $N_s[2]$  and finds a match. It concludes that the VU is a generation-2 VU, and therefore sends back  $K_P[2]$ .

### 12.3. VU – Motion Sensor Pairing and Communication using AES

TCS\_360

As specified in Table 45 in section 9.2.1, all keys involved in the pairing of a (second-generation) vehicle unit and a motion sensor and in subsequent communication shall be AES keys, rather than double-length TDES keys as specified in [ISO 16844-3]. These AES keys may have a length of 128, 192 or 256 bits. Since the AES block size is 16 bytes, the length of an encrypted message must be a multiple of 16 bytes, compared to 8 bytes for TDES. Moreover, some of these messages will be used to transport AES keys, the length of which may be 128, 192 or 256 bits. Therefore, the number of data bytes per instruction in Table 5 of [ISO 16844-3] shall be changed as shown in Table 48:

<i>Instruction</i>	<i>Request / reply</i>	<i>Description of data</i>	<i># of plaintext data bytes according to</i>	<i># of plaintext data bytes</i>	<i># of encrypted data bytes when using AES keys of bitlength</i>
--------------------	------------------------	----------------------------	---	----------------------------------	---

<sup>12</sup> Note that the generation-1, generation-2 and generation-3 pairing keys may actually be the same key, or may be three different keys having different lengths, as explained in TCS\_259.



			[ISO 16844-3]	using AES keys	128	192	256
10	request	Authentication data + file number	8	8	16	16	16
11	reply	Authentication data + file contents	16 or 32, depend on file	16 or 32, depend on file	16 / 32	16 / 32	16 / 32
41	request	MoS serial number	8	8	16	16	16
41	reply	Pairing key	16	16 / 24 / 32	16	32	32
42	request	Session key	16	16 / 24 / 32	16	32	32
43	request	Pairing information	24	24	32	32	32
50	reply	Pairing information	24	24	32	32	32
70	request	Authentication data	8	8	16	16	16
80	reply	MoS counter value + auth. data	8	8	16	16	16

**Table 48 Number of plaintext and encrypted data bytes per instruction defined in [ISO 16844-3]**

TCS\_361 The pairing information that is sent in instructions 43 (VU request) and 50 (MoS reply) shall be assembled as specified in section 7.6.10 of [ISO 16844-3], except that the AES algorithm shall be used instead of the TDES algorithm in the pairing data encryption scheme, thus resulting in two AES encryptions, and adopting the padding specified in TCS\_362 to fit with the AES block size. The key  $K'_p$  used for this encryption shall be generated as follows:

- In case the pairing key  $K_P$  is 16 bytes long:  $K'_p = K_P \text{ XOR } (N_s || N_s)$
- In case the pairing key  $K_P$  is 24 bytes long:  $K'_p = K_P \text{ XOR } (N_s || N_s || N_s)$
- In case the pairing key  $K_P$  is 32 bytes long:  $K'_p = K_P \text{ XOR } (N_s || N_s || N_s || N_s)$

where  $N_s$  is the 8-byte serial number of the motion sensor.

TCS\_362 In case the plaintext data length (using AES keys) is not a multiple of 16 bytes, padding method 2 defined in [ISO 9797-1] shall be used.

Note: in [ISO 16844-3], the number of plaintext data bytes is always a multiple of 8, such that padding is not necessary when using TDES. The definition of data and messages in [ISO 16844-3] is not changed by this part of this Appendix, thus necessitating the application of padding.

TCS\_363

For instruction 11 and in case more than one block of data must be encrypted, the Cipher Block Chaining mode of operation shall be used as defined in [ISO 10116], with an interleave parameter  $m = 1$ . The IV to be used shall be

- For instruction 11: the 8-byte authentication block specified in section 7.6.3.3 of [ISO 16844-3], padded using padding method 2 defined in [ISO 9797-1]; see also section 7.6.5 and 7.6.6 of [ISO 16844-3].
- For all other instructions in which more than 16 bytes are transferred, as specified in Table 48: '00' {16}, i.e. sixteen bytes with binary value 0.

Note: As shown in section 7.6.5 and 7.6.6 of [ISO 16844-3], when the MoS encrypts data files for inclusion in instruction 11, the authentication block is both

- Used as the initialization vector for the CBC-mode encryption of the data files
- Encrypted and included as the first block in the data that is sent to the VU.

#### 12.4. VU – Motion Sensor Pairing For Different Equipment Generations

TCS\_364

As explained in section 9.2.1, a second-generation motion sensor may contain the TDES-based encryption of the pairing data (as defined in Part A of this Appendix), which allows the motion sensor to be paired to a first-generation VU. If this is the case, a first-generation VU and a second-generation motion sensor shall be paired as described in Part A of this Appendix and in [ISO 16844-3]. For the pairing process either a first-generation or a second-generation workshop card may be used.

Notes:

- It is not possible to pair a second-generation VU to a first-generation motion sensor.
- It is not possible to use a first-generation workshop card for coupling a second-generation VU to a motion sensor.

### 13. Security for Remote Communication over DSRC

#### 13.1. General

As specified in Appendix 14, a VU regularly generates Remote Tachograph Monitoring (RTM) data and sends this data to the (internal or external) Remote Communication Facility (RCF). The remote communication facility is responsible for sending this data over the DSRC interface described in Appendix 14 to the remote interrogator. Appendix 1 specifies that the RTM data is the concatenation of:

- **Encrypted tachograph payload** the encryption of the plaintext tachograph payload
- **DSRC security data** described below

The plaintext tachograph payload data format is specified in Appendix 1 and further described in Appendix 14. This section describes the structure of the DSRC security data; the formal specification is in Appendix 1.

TCS\_365 The plaintext tachographPayload data communicated by a VU to a Remote Communication Facility (if the RCF is external to the VU) or from the VU to a remote interrogator over the DSRC interface (if the RCF is internal in the VU) shall be protected in encrypt-then-authenticate mode, i.e. the tachograph payload data is encrypted first to ensure message confidentiality, and afterwards a MAC is calculated to ensure data authenticity and integrity.

TCS\_366 The DSRC security data shall consist of the concatenation of the following data elements in the following order; see also Figure 12:

- **Current date time** the current date and time of the VU (data type TimeReal)
- **Counter** a 3-byte counter, see TCS\_367
- **VU serial number** the VU's serial number (data type VuSerialNumber)
- **DSRC master key version number** the 1-byte version number of the DSRC master key from which the VU-specific DSRC keys were derived, see section 9.2.3.
- **MAC** the MAC calculated over all previous bytes in the RTM data.

TCS\_367 The 3-byte counter in the DSRC security data shall be in MSB-first format. The first time a VU calculates a set of RTM data after it is taken into production, it shall set the value of the counter to 0. The VU shall increase the value of the counter data by 1, each time before it calculates a next set of RTM data.

13.2. Tachograph Payload Encryption and MAC Generation

TCS\_368 Given a plaintext data element with data type `TachographPayload` as described in Appendix 14, a VU shall encrypt this data as shown in Figure 12: the VU's DSRC key for encryption  $K_{VU_{DSRC\_ENC}}$  (see section 9.2.3) shall be used with AES in the Cipher Block Chaining (CBC) mode of operation, as defined in [ISO 10116], with an interleave parameter  $m = 1$ . The initialization vector shall be equal to  $IV = current\ date\ time \parallel '00\ 00\ 00\ 00\ 00\ 00\ 00\ 00' \parallel counter$ , where *current date time* and *counter* are specified in TCS\_366. The data to be encrypted shall be padded using method 2 defined in [ISO 9797-1].

TCS\_369 A VU shall calculate the MAC in the DSRC security data as shown in Figure 12: the MAC shall be calculated over all preceding bytes in the RTM data, up to and including the DSRC master key version number, and including the tags and lengths of the data objects. The VU shall use its DSRC key for authenticity  $K_{VU_{DSRC\_MAC}}$  (see section 9.2.3) with the AES algorithm in CMAC mode as specified in [SP 800-38B]. The length of the MAC shall be linked to the length of the VU-specific DSRC keys, as specified in TCS\_192.

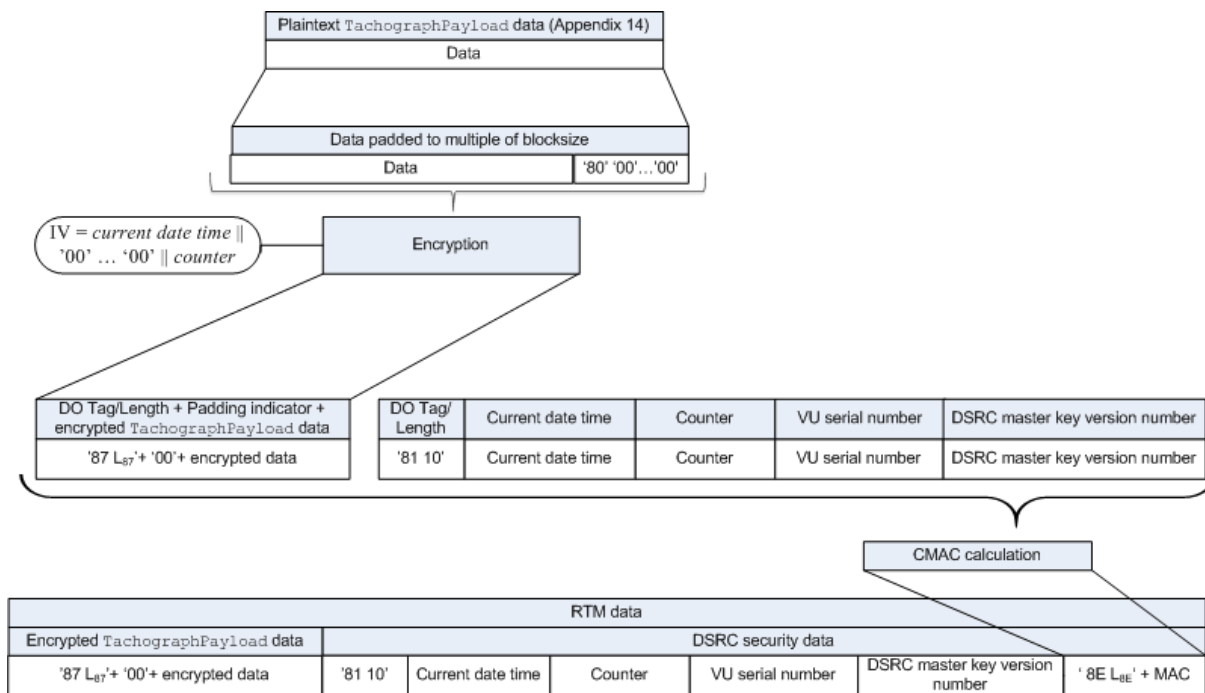


Figure 12 Tachograph payload encryption and MAC generation

13.3. Verification and Decryption of Tachograph Payload

TCS\_370 When a remote interrogator receives RTM data from a VU, it shall send the entire RTM data to a control card in the data field of a PROCESS DSRC MESSAGE command, as described in Appendix 2. Then:

1. The control card shall inspect the DSRC master key version number in the DSRC security data. If the control card does not know the indicated DSRC master key, it shall return an error specified in Appendix 2 and abort the process.

2. The control card shall use the indicated DSRC master key in combination with the VU serial number in the DSRC security data to derive the VU-specific DSRC keys  $K_{VU_{DSRC\_ENC}}$  and  $K_{VU_{DSRC\_MAC}}$ , as specified in TCS\_266.
3. The control card shall use  $K_{VU_{DSRC\_MAC}}$  to verify the MAC in the DSRC security data, as specified in TCS\_369. If the MAC is incorrect, the control card shall return an error specified in Appendix 2 and abort the process.
4. The control card shall use  $K_{VU_{DSRC\_ENC}}$  to decrypt the encrypted tachograph payload, as specified in TCS\_368. The control card shall remove the padding and shall return the decrypted tachograph payload data to the remote interrogator.

TCS\_371 In order to prevent replay attacks, the remote interrogator shall verify the freshness of the RTM data by verifying that the *current date time* in the DSRC security data does not deviate too much from the current time of the remote interrogator.

Notes:

- This requires the remote interrogator to have an accurate and reliable source of time.
- Since Appendix 14 requires a VU to calculate a new set of RTM data every 60 seconds, and the clock of the VU is allowed to deviate 1 minute from the real time, a lower limit for the freshness of the RTM data is 2 minutes. The actual freshness to be required also depends on the accuracy of the clock of the remote interrogator.

TCS\_372 When a workshop verifies the correct functioning of the DSRC functionality of a VU, it shall send the entire RTM data received from the VU to a workshop card in the data field of a PROCESS DSRC MESSAGE command, as described in Appendix 2. The workshop card shall perform all checks and actions specified in TCS\_370.

## 14. Signing Data Downloads and Verifying Signatures

### 14.1. General

TCS\_373 The Intelligent Dedicated Equipment (IDE) shall store data received from a VU or a card during one download session within one physical data file. Data may be stored on an ESM (external storage medium). This file contains digital signatures over data blocks, as specified in Appendix 7. This file shall also contain the following certificates (refer to section 9.1):

- In case of a VU download:
  - o The VU\_Sign certificate
  - o The MSCA\_VU-EGF certificate containing the public key to be used for verification of the VU\_Sign certificate
- In case of a Card download:
  - o The Card\_Sign certificate
  - o The MSCA\_Card certificate containing the public key to be used for verification of the Card\_Sign certificate

TCS\_374 The IDE shall also dispose of.

- In case it uses a control card to verify the signature, as shown in Figure 13: The link certificate linking the latest EUR certificate to the EUR certificate whose validity period directly precedes it, if existing.
- In case it verifies the signature itself: all valid European root certificates.

Note: the method the IDE uses to retrieve these certificates is not specified in this Appendix.

### 14.2. Signature generation

TCS\_375 The signing algorithm to create digital signatures over downloaded data shall be ECDSA as specified in [DSS], using the hashing algorithm linked to the key size of the VU or the card, as specified in TCS\_192. The signature format shall be plain, as specified in [TR-03111].

### 14.3. Signature verification

TCS\_376 An IDE may perform verification of a signature over downloaded data itself or it may use a control card for this purpose. In case it uses a control card, signature verification shall take place as shown in Figure 13. In case it performs signature verification itself, the IDE shall verify the authenticity and validity of all certificates in the certificate chain in the data file, and it shall verify the signature over the data following the signature scheme defined in [DSS].

Notes to Figure 13 :

- The equipment that signed the data to be analysed is denoted EQT.
- The EQT certificates and public keys mentioned in the figure are those for signing, i.e. VU\_Sign or Card\_Sign.

- The EQT.CA certificates and public keys mentioned in the figure are those for signing VU or Card certificates, as applicable.
- The EQT.CA.EUR certificate mentioned in the figure is the European root certificate that is indicated in the CAR of the EQT.CA certificate.
- The EQT.Link certificate mentioned in the figure is the EQT's link certificate, if present. As specified in section 9.1.2, this is a link certificate for a new European root key pair created by the ERCA and signed with the previous European private key.
- The EQT.Link.EUR certificate is the European root certificate that is indicated in the CAR of the EQT.Link certificate.

TCS\_377 For calculating the hash M sent to the control card in the PSO:Hash command, the IDE shall use the hashing algorithm linked to the key size of the VU or the card from which the data is downloaded, as specified in TCS\_192.

TCS\_378 For verifying the EQT's signature, the control card shall follow the signature scheme defined in [DSS].

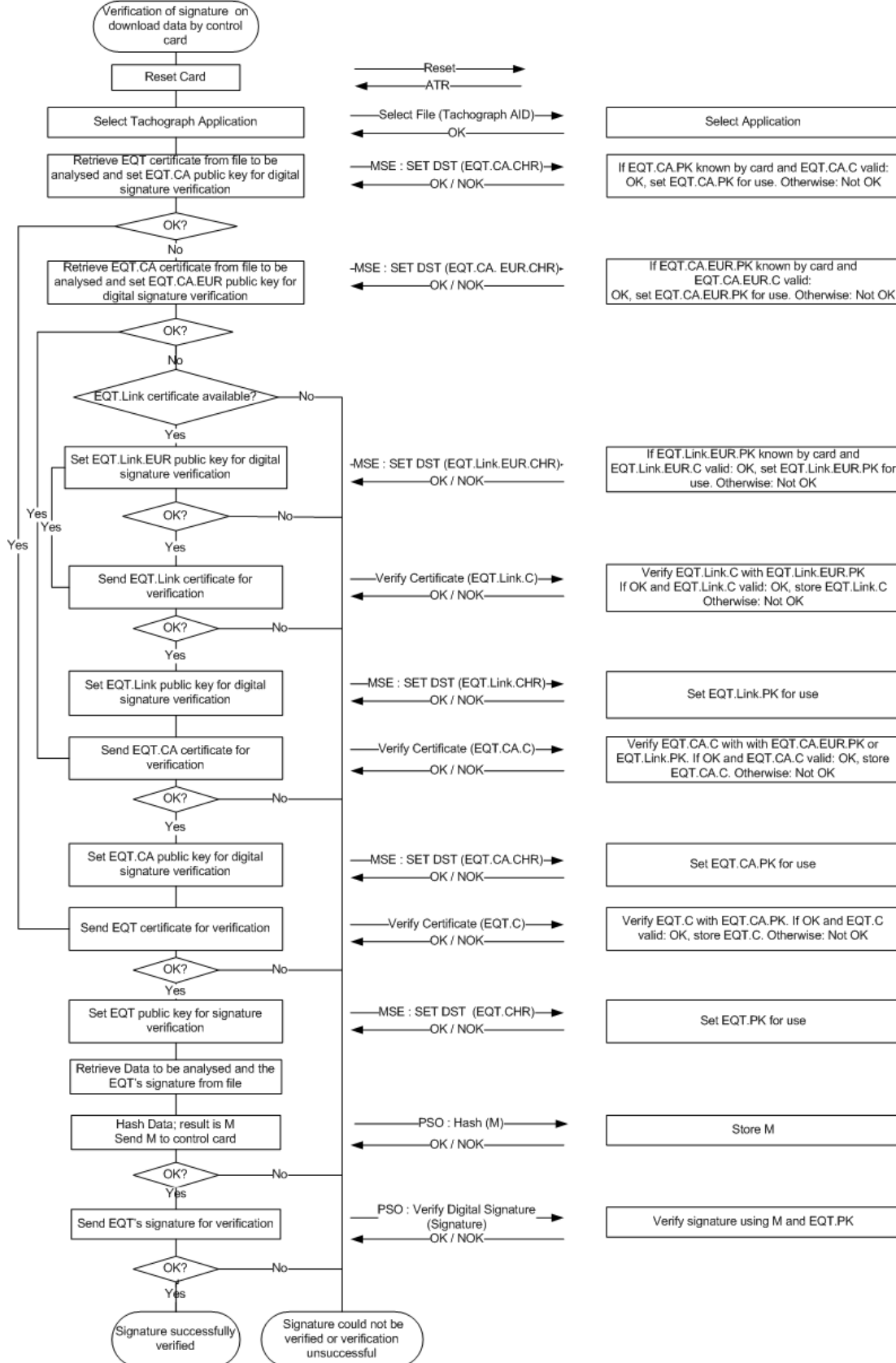
Note: This document does not specify any action to undertake if a signature over a downloaded data file cannot be verified or if the verification is unsuccessful.

**Figure 13 Protocol for verification of the signature over a downloaded data file**



ESM / IDE

CONTROL CARD



## APPENDIX 12 POSITIONING BASED ON GLOBAL NAVIGATION SATELLITE SYSTEM (GNSS)

### TABLE OF CONTENT

#### Contents

<b>APPENDIX 12 POSITIONING BASED ON GLOBAL NAVIGATION SATELLITE SYSTEM (GNSS)</b>	<b>498</b>
<b>1. INTRODUCTION</b>	<b>500</b>
<b>1.1. Scope</b>	<b>500</b>
<b>1.2. Acronyms and notations</b>	<b>501</b>
<b>2. SPECIFICATION OF THE GNSS RECEIVER</b>	<b>501</b>
<b>3. NMEA SENTENCES</b>	<b>502</b>
<b>4. VEHICLE UNIT WITH AN EXTERNAL GNSS FACILITY</b>	<b>503</b>
<b>4.1. Configuration</b>	<b>503</b>
4.1.1 .. Main components and interfaces	503
4.1.2 .. External GNSS facility state at the end of production	504
<b>4.2. Communication between the external GNSS facility and the vehicle unit</b>	<b>505</b>
4.2.1 .. Communication Protocol	505
4.2.2 .. Secure transfer of GNSS data	507
4.2.3 .. Structure of the Read Record command	508
<b>4.3. Coupling, mutual authentication and session key agreement of the external GNSS facility with vehicle unit</b>	<b>509</b>
<b>4.4. Error Handling</b>	<b>509</b>
4.4.1 .. Communication error with the external GNSS facility	509
4.4.2 .. Breach of the physical integrity of the external GNSS facility	509
4.4.3 .. Absence of position information from GNSS receiver	509
4.4.4 .. External GNSS facility certificate expired	510
<b>5. VEHICLE UNIT WITHOUT AN EXTERNAL GNSS FACILITY</b>	<b>510</b>
<b>5.1. Configuration</b>	<b>510</b>
<b>5.2. Error Handling</b>	<b>510</b>

5.2.1 .. Absence of position information from GNSS receiver	510
<b>6. GNSS TIME CONFLICT</b> _____	<b>511</b>
<b>7. VEHICLE MOTION CONFLICT</b> _____	<b>511</b>

1. Introduction

This Appendix provides the technical requirements for the GNSS data used by the Vehicle Unit, including the protocols that must be implemented to assure the secure and correct data transfer of the positioning information.

The main articles in this Regulation (EU) N°. 165/2014 driving these requirements are: “Article 8 Recording of the position of the vehicle at certain points during the daily working period”, “Article 10 Interface with Intelligent Transport Systems” and “Article 11 Detailed provisions for smart tachographs”.

1.1. Scope

GNS\_1

The Vehicle Unit shall collect location data from at least one GNSS to support the implementation of Article 8.

The Vehicle Unit may be with or without an external GNSS facility as described in Figure 14:

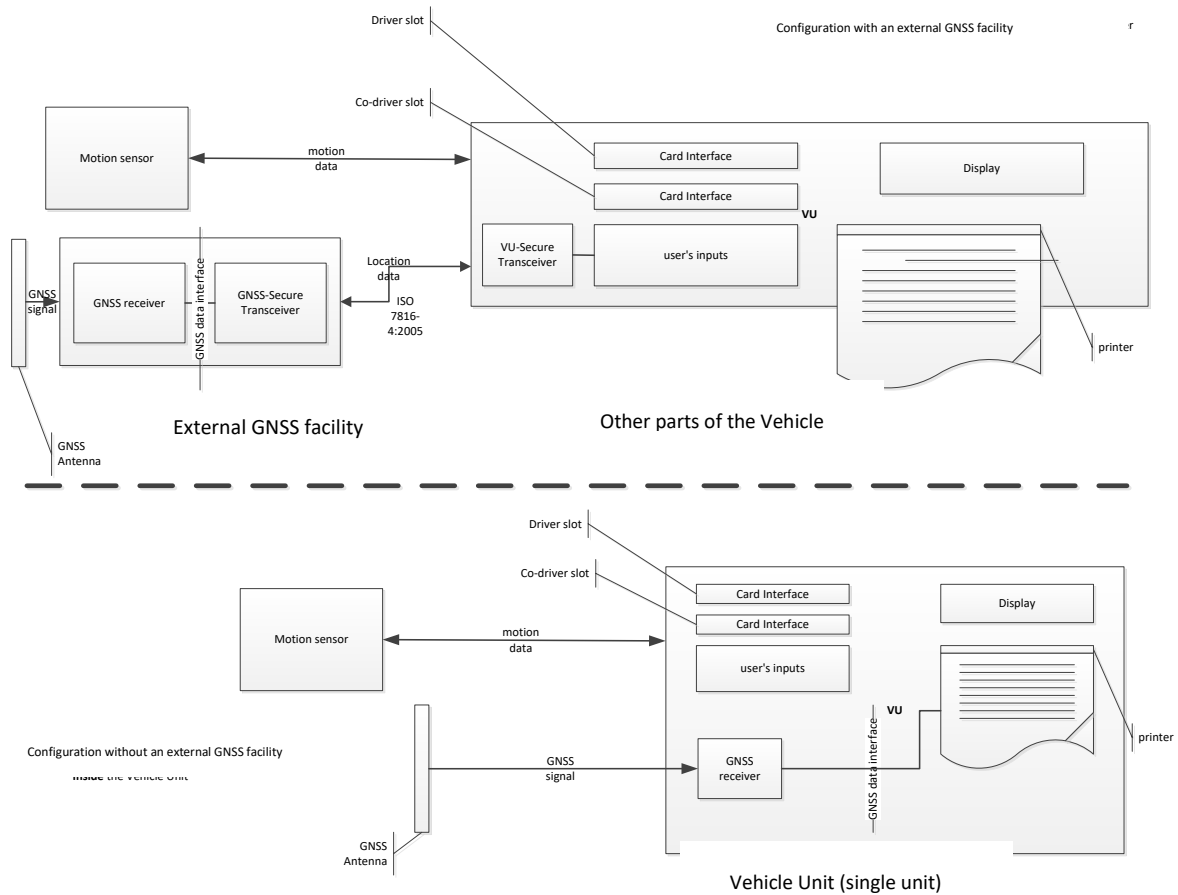


Figure 14 Different configurations for GNSS receiver.

## 1.2. Acronyms and notations

The following acronyms are used in this appendix:

DOP	Dilution of Precision
EGF	Elementary file GNSS Facility
EGNOS	European Geostationary Navigation Overlay Service
GNSS	Global Navigation Satellite System
GSA	GPS DOP and active satellites
HDOP	Horizontal Dilution of Precision
ICD	Interface Control Document
NMEA	National Marine Electronics Association
PDOP	Position Dilution of Precision
RMC	Recommended Minimum Specific
SIS	Signal in Space
VDOP	Vertical Dilution of Precision
VU	Vehicle Unit

## 2. Specification of the GNSS receiver

Regardless of the configuration of the Smart Tachograph with or without an external GNSS facility, the provision of accurate and reliable positioning information is an essential element of the effective operation of the Smart Tachograph. Therefore, it is appropriate to require its compatibility with the services provided by the Galileo and European Geostationary Navigation Overlay Service (EGNOS) programmes as set out in Regulation (EU) N<sup>o</sup>. 1285/2013 of the European Parliament and of the Council<sup>13</sup>. The system established under the Galileo programme is an independent global satellite navigation system and the one established under the EGNOS programme is a regional satellite navigation system improving the quality of the Global Positioning System signal.

GNS\_2 Manufacturers shall ensure that the GNSS receivers in the Smart Tachographs are compatible with the positioning services provided by the Galileo and the EGNOS systems. Manufacturers may also choose, in addition, compatibility with other satellite navigation systems.

GNS\_3 The GNSS receiver shall have the capability to support Authentication on the Open Service of Galileo when such service will be provided by the Galileo system and supported by GNSS receiver manufacturers. However, for smart tachographs introduced in the market before the previous conditions are satisfied and not having the capability to support Authentication of the Open Service of Galileo, no retrofitting will be required.

<sup>13</sup> Regulation (EU) N<sup>o</sup>. 1285/2013 of the European Parliament and of the Council of 11 December 2013 on the implementation and exploitation of European satellite navigation systems and repealing Council Regulation (EC) N<sup>o</sup>. 876/2002 and Regulation (EC) N<sup>o</sup>. 683/2008 of the European Parliament and of the Council (OJ L 347, 20.12.2013, p. 1).

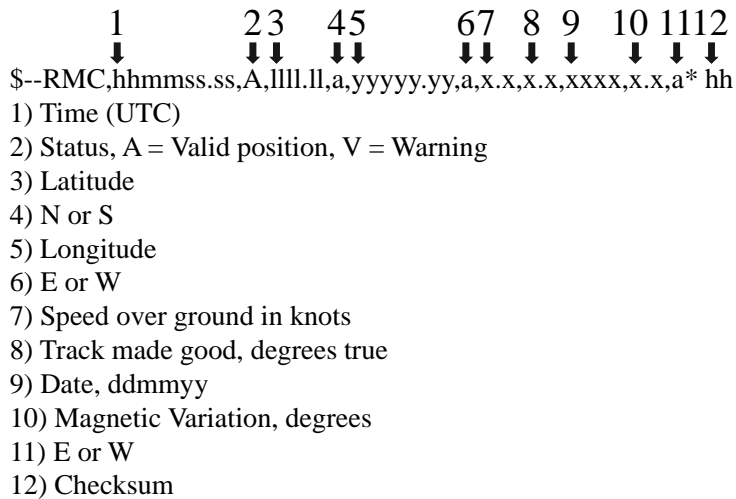
### 3. NMEA Sentences

This section describes the NMEA sentences used in the functioning of the Smart Tachograph. This section is valid both for the configuration of the Smart Tachograph with or without an external GNSS facility.

GNS\_4

The location data is based on the NMEA sentence Recommended Minimum Specific (RMC) GNSS Data, which contains the Position information (Latitude, Longitude), Time in UTC format (hhmmss.ss), and Speed Over Ground in Knots plus additional values.

The format of the RMC sentence is the following (as from NMEA V4.1 standard):



**Figure 15 Structure of the RMC sentence**

The Status gives indication if the GNSS signal is available. Until the value of the Status is not set to A, the received data (e.g., on Time or Latitude/Longitude) cannot be used to record the position of the vehicle in the VU.

The resolution of the position is based on the format of the RMC sentence described above. The first part of the fields 3) and 5) (the first two numbers) are used to represent the degrees. The rest are used to represent the minutes with three decimals. So the resolution is 1/1000 of minute or 1/60000 of degree (because one minute is 1/60 of a degree).

GNS\_5

The Vehicle Unit shall store in the VU database the position information for latitude and longitude with a resolution of 1/10 of minute or 1/600 of a degree as described in Appendix 1 for type GeoCoordinates.

The GPS DOP and active satellites (GSA) command can be used by the VU to determine and record the signal availability and accuracy. In particular the HDOP is used to provide an indication on the level of accuracy of the



- GNS\_9 The external GNSS facility shall consist of the following components (see Figure 17):
- a) A commercial GNSS receiver to provide the position data through the GNSS data interface. For example, the GNSS data interface can be NMEA standard V4.10 where The GNSS receiver acts as a talker and transmit NMEA sentences to the GNSS Secure Transceiver with a frequency of 1Hz for the pre-defined set of NMEA sentences, which must include at least the RMC and GSA sentences. The implementation of the GNSS data interface is a choice of the manufacturers of the external GNSS facility.
  - b) A transceiver unit (GNSS Secure Transceiver) with the capability to support standard ISO/IEC 7816-4:2013 (see 5.2.1) to communicate with the vehicle unit and support the GNSS data interface to the GNSS receiver. The unit is provided with a memory to store the identification data of the GNSS receiver and external GNSS facility.
  - c) An enclosure system with tamper detection function, which encapsulate both the GNSS receiver and the GNSS Secure Transceiver. The tamper detection function shall implement the security protection measures as requested in the Protection Profile of the Smart Tachograph.
  - d) A GNSS antenna installed on the vehicle and connected to the GNSS receiver through the enclosure system.
- GNS\_10 The external GNSS facility has at least the following external interfaces:
- a) the interface to the GNSS antenna installed on the vehicle truck, if an external antenna is used.
  - b) the interface to the Vehicle Unit.
- GNS\_11 In the VU, the VU Secure Transceiver is the other end of the secure communication with the GNSS Secure Transceiver and it must support ISO/IEC 7816-4:2013 for the connection to the external GNSS facility.
- GNS\_12 For the physical layer of the communication with the external GNSS facility, the vehicle unit shall support ISO/IEC 7816-12:2005 or another standard able to support ISO/IEC 7816-4:2013. (see 5.2.1).
- 5.1.1 External GNSS facility state at the end of production
- GNS\_13 The external GNSS facility shall store the following values in the non-volatile memory of the GNSS Secure Transceiver when it leaves the factory:
- the EGF\_MA key pair and corresponding certificate,
  - the MSCA\_VU-EGF certificate containing the MSCA\_VU-EGF.PK public key to be used for verification of the EGF\_MA certificate,
  - the EUR certificate containing the EUR.PK public key to be used for verification of the MSCA\_VU-EGF certificate,
  - the EUR certificate whose validity period directly precedes the validity period of the EUR certificate to be used to verify the MSCA\_VU-EGF certificate, if existing,
  - the link certificate linking these two EUR certificates, if existing,
  - the extended serial-number of the external GNSS facility,
  - operating system identifier of the GNSS facility,
  - type approval number of the external GNSS facility;



- Identifier of the security component of the external GNSS module.

## 5.2. Communication between the external GNSS facility and the vehicle unit

### 5.2.1 Communication Protocol

- GNS\_14 The communication protocol between the external GNSS facility and the vehicle unit shall support three functions:
1. The collection and distribution of GNSS data (e.g., position, timing, speed),
  2. The collection of the configuration data of the external GNSS facility,
  3. The management protocol to support the coupling, mutual authentication and session key agreement between the external GNSS facility and the VU.
- GNS\_15 The communication protocol shall be based on standard ISO/IEC 7816-4:2013 with the VU Secure Transceiver playing the master role and the GNSS Secure Transceiver playing the slave role. The physical connection between the external GNSS facility and the vehicle unit is based on ISO/IEC 7816-12:2005 or another standard able to support ISO/IEC 7816-4:2013
- GNS\_16 In the communication protocol, extended length fields shall not supported.
- GNS\_17 The communication protocol of ISO 7816 (both \*-4:2013 and \*-12:2005) between the external GNSS facility and the VU shall be set to T=1.
- GNS\_18 Regarding the functions 1) the collection and distribution of GNSS data and 2) the collection of the configuration data of the external GNSS facility and 3) management protocol, the GNSS Secure Transceiver shall simulate a smart card with a file system architecture composed by a Master File (MF), a Directory File (DF) with Application Identifier specified in Appendix 1 chapter 6.2 (' FF 44 54 45 47 4D') and with 3 EFs containing certificates and one single Elementary File (EF.EGF) with file identifier equal to '2F2F' as described in Table 49.
- GNS\_19 The GNSS Secure Transceiver shall store the data coming from the GNSS receiver and the configuration in the EF.EGF. This is a linear, variable-length record file with an identifier equal to '2F2F' in hexadecimal format.
- GNS\_20 The GNSS Secure Transceiver shall use a memory to store the data able to perform at least 20 millions write/read cycles. Apart from this aspect, the internal design and implementation of the GNSS Secure Transceiver is left to the manufacturers.
- The mapping of record numbers and data is provided in Table 49. Note that there are four GSA sentences for the four satellite systems and Satellite-Based Augmentation System (SBAS).
- GNS\_21 The file structure is provided in Table 49. For the access conditions (ALW, NEV, SM-MAC) see Appendix 2 chapter 3.5.

Table 49 File Structure

					<i>Access conditions</i>		
File	File ID	Read	Update	Encrypted			
MF	3F00						
EF.ICC	0002	ALW	NEV (by VU)	No			
DF GNSS Facility	0501	ALW	NEV	No			
EF EGF_MACertificate	C100	ALW	NEV	No			
EF CA_Certificate	C108	ALW	NEV	No			
EF Link_Certificate	C109	ALW	NEV	No			
EF.EGF	2F2F	SM- MAC	NEV (by VU)	No			
<i>File / Data element</i>		<i>Record no</i>	<i>Size (bytes)</i>		<i>Default values</i>		
			<b>Min</b>	<b>Max</b>			
MF			552	1031			
EF.ICC							
sensorGNSSSerialNumber			8	8			
DF GNSS Facility			612	1023			
EF EGF_MACertificate			204	341			
EGFCertificate			204	341	{00..00}		
EF CA_Certificate			204	341			
MemberStateCertificate			204	341	{00..00}		
EF Link_Certificate			204	341			
LinkCertificate			204	341	{00..00}		
EF.EGF							
RMC NMEA Sentence	'01'		85	85			
1st GSA NMEA Sentence	'02'		85	85			
2nd GSA NMEA Sentence	'03'		85	85			
3rd GSA NMEA Sentence	'04'		85	85			
4th GSA NMEA Sentence	'05'		85	85			
5th GSA NMEA Sentence	'06'		85	85			
Extended serial-number of the external GNSS facility defined in Appendix 1 as	'07'		8	8			

<i>File / Data element</i>	<i>Record no</i>	<i>Size (bytes)</i>	<i>Default values</i>
<b>SensorGNSSSerialNumber.</b>			
Operating system identifier of the GNSS			
Secure Transceiver defined in Appendix 1 as SensorOSIdentifier.	'08'	2	2
Type approval number of the external GNSS facility defined in Appendix 1 as			
SensorExternalGNSSApprovalNumber.	'09'	16	16
Identifier of the security component of the external GNSS facility defined in Appendix 1 as			
SensorExternalGNSSIdentifier	'10'	8	8
RFU – Reserved for Future Use	From '11' to 'FD'		

### 5.2.2 Secure transfer of GNSS data

GNS\_22 The secure transfer of GNSS position data shall be allowed only in the following conditions:

1. The coupling process has been completed as described in Appendix 11. Common security mechanisms.
2. The periodic mutual authentication and session key agreement between the VU and the external GNSS facility also described in Appendix 11. Common security mechanisms has been executed with the indicated frequency.

GNS\_23 Every T seconds, where T is a value lower or equal to 10, unless coupling or mutual authentication and session key agreement takes place, the VU requests from the external GNSS facility the position information on the basis of the following flow:

1. The VU requests location data from the External GNSS facility together with Dilution of Precision data (from the GSA NMEA sentence). The VU Secure Transceiver shall use the ISO/IEC 7816-4:2013 SELECT and READ RECORD(S) command in secure messaging authentication-only mode as described in Appendix 11 section 11.5 with the file identifier “2F2F” and RECORD number equal to “01” for RMC NMEA sentence and ‘02’, ‘03’, ‘04’, ‘05’, ‘06’ for GSA NMEA sentence.
2. The last location data received is stored in the EF with identifier ‘2F2F’ and the records described in Table 49 in the GNSS secure transceiver as the GNSS secure transceiver receives NMEA data with a frequency of at least 1 Hz from the GNSS receiver through the GNSS data interface.
3. The GNSS Secure Transceiver sends the response to the VU Secure Transceiver by using the APDU response message in secure messaging authentication-only mode as described in Appendix 11 section 11.5.
4. The VU Secure Transceiver checks the authenticity and integrity of the received response. In case of positive outcome, the location data is transferred to the VU processor through the GNSS data interface.
5. The VU processor checks the received data extracting the information (e.g., latitude, longitude, time) from the RMC NMEA sentence. The RMC NMEA sentence includes the information if the position is valid. If the position is not valid, the location data is not available yet and it cannot be used to record the position of the

vehicle. If the position is valid, the VU processor also extracts the values of HDOP from GSA NMEA sentences and calculate the average value on the available satellite systems (i.e., when the fix is available).

6. The VU processor stores the received and processed information such as latitude, longitude, time and speed in the VU in the format defined in Appendix 1 Data Dictionary as GeoCoordinates together with the value of HDOP calculated as the minimum of the HDOP values collected on the available GNSS systems.

### 5.2.3 Structure of the Read Record command

This section describes in detail the structure of the Read Record command. Secure messaging (authentication-only mode) is added as described in Appendix 11 Common security mechanisms.

GNS\_24 The command shall support the Secure Messaging authentication-only-mode, see Appendix 11.

GNS\_25 Command Message

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
CLA	1	'0Ch'	Secure messaging asked.
INS	1	'B2h'	Read Record
P1	1	'XXh'	Record number ('00' references the current record)
P2	1	'04h'	Read the record with the record number indicated in P1
Le	1	'XXh'	Length of data expected. Number of Bytes to be read.

GNS\_26 The record referenced in P1 becomes the current record.

<i>Byte</i>	<i>Length</i>	<i>Value</i>	<i>Description</i>
#1-#X	X	'XX..XXh'	Data read
SW	2	'XXXXh'	Status Words (SW1,SW2)

- If the command is successful, the GNSS secure transceiver returns '**9000**'.
- If the current file is not record oriented, the GNSS secure transceiver returns '**6981**'.
- If the command is used with P1 = '00' but there is no current EF the GNSS secure transceiver returns '**6986**' (command not allowed).
- If the record is not found, the GNSS secure transceiver returns '**6A 83**'.
- If the external GNSS facility has detected tampering, it shall return status words '**66 90**'.

GNS\_27 The GNSS Secure Transceiver shall support the following tachograph generation 2 commands specified in Appendix 2:

<i>Command</i>	<i>Reference</i>
Select	Appendix 2 chapter 3.5.1
Read Binary	Appendix 2 chapter 3.5.2
Get Challenge	Appendix 2 chapter 3.5.4
PSO: Verify Certificate	Appendix 2 chapter 3.5.7
External Authenticate	Appendix 2 chapter 3.5.9
General Authenticate	Appendix 2 chapter 3.5.10
MSE:SET	Appendix 2 chapter 3.5.11

### 5.3. Coupling, mutual authentication and session key agreement of the external GNSS facility with vehicle unit

The coupling, mutual authentication and session key agreement of the external GNSS facility with the vehicle unit is described in Appendix 11. Common security mechanisms, Chapter 11.

### 5.4. Error Handling

This section describes how potential error conditions by the external GNSS facility are addressed and recorded in the VU.

#### 5.4.1 Communication error with the external GNSS facility

GNS\_28 If the VU does not manage to communicate to the coupled external GNSS facility for more than 20 continuous minutes, the VU shall generate and record in the VU an event of type `EventFaultType` with the value of enum `'53'H External GNSS communication fault` and with the timestamp set to the current time. The event will be generated only if the following two conditions are satisfied: a) the Smart Tachograph is not in calibration mode and b) the vehicle is moving. In this context, a communication error is triggered when the VU Secure Transceiver does not receive a response message after a request message as described in 5.2.

#### 5.4.2 Breach of the physical integrity of the external GNSS facility

GNS\_29 If the external GNSS facility has been breached, the GNSS Secure Transceiver shall erase all its memory including cryptographic material. As described in GNS\_25 and GNS\_26, the VU shall detect tampering if the Response has status `'6690'`. The VU shall then generate an event of type `EventFaultType` enum `'55'H Tamper detection of GNSS`.

#### 5.4.3 Absence of position information from GNSS receiver

GNS\_30 If the GNSS Secure Transceiver does not receive data from the GNSS receiver for more than 3 continuous hours, the GNSS Secure Transceiver shall generate a response message to the READ RECORD command with RECORD number equal to `'01'` with a Data Field of 12 bytes all set to `0xFF`. Upon reception of the Response message with this value of the data field, the VU shall generate and record an event of type `EventFaultType` enum `'52'H external GNSS receiver fault` with a timestamp equal to the current value of time only if the following two conditions are satisfied: a) the Smart Tachograph is not in calibration mode and b) the vehicle is moving.

5.4.4 External GNSS facility certificate expired

GNS\_31 If the VU detects that the EGF certificate used for mutual authentication is not valid any longer, the VU shall generate and record a recording equipment fault of typeEventFaultType enum '56'H External GNSS facility certificate expired with a timestamp equal to the current value of time. The VU shall still use the received GNSS position data.

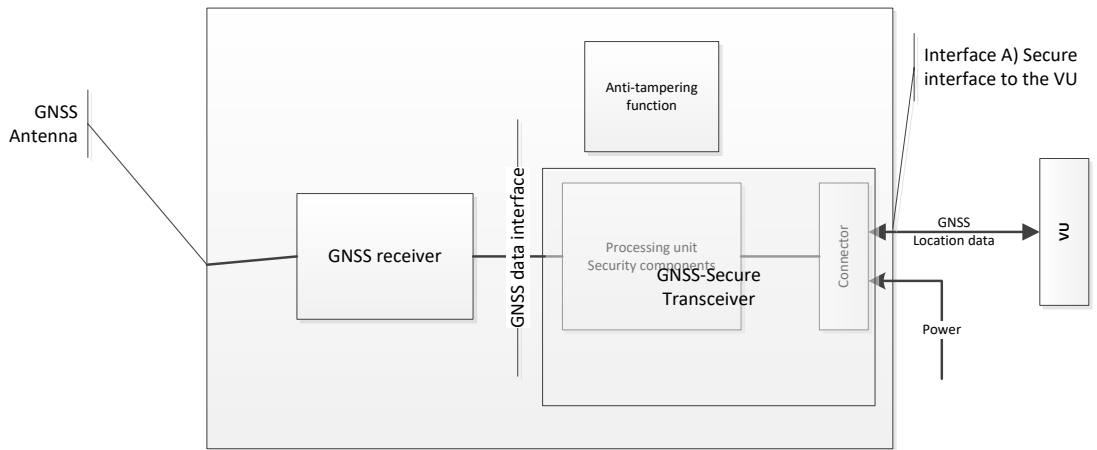


Figure 17 Schema of the external GNSS facility

6. Vehicle Unit without an external GNSS facility

6.1. Configuration

In this configuration, the GNSS receiver is inside the Vehicle Unit as described in Figure 14.

GNS\_32 The GNSS receiver shall act as a talker and transmit NMEA sentences to the VU processor, which shall act as a listener with a frequency of 1/10 Hz or faster for the pre-defined set of NMEA sentences, which shall include at least the RMC and GSA sentences.

GNS\_33 An external GNSS antenna installed on the vehicle or an internal GNSS antenna shall be connected to the VU.

6.2. Error Handling

6.2.1 Absence of position information from GNSS receiver

GNS\_34 If the VU does not receive data from the GNSS receiver for more than 3 continuous hours, the VU shall generate and record an event of type EventFaultType enum '51'H Internal GNSS receiver fault with a timestamp equal to the current value of time only if the following two conditions are satisfied: a) the Smart Tachograph is not in calibration mode and b) the vehicle is moving.

#### 7. GNSS Time Conflict

If the VU detects a discrepancy of more than 1 minute between the time of the vehicle unit's time measurement function and the time originating from the GNSS receiver, the VU will record an event of type EventFaultType enum *'OB'H Time conflict (GNSS versus VU internal clock)*. This event is recorded together with the internal clock value of the vehicle unit and comes together with an automatic time adjustment. After a time conflict event has been triggered, the VU will not check the time discrepancy for the next 12 hours. This event shall not be triggered in cases no valid GNSS signal was detectable by the GNSS receiver within the last 30 days. However, when the position information from the GNSS receiver is available again, the automatic time adjustment shall be done.

#### 8. Vehicle motion conflict

GNS\_35 The VU shall trigger and record an Vehicle Motion Conflict event (see in requirement 84 in this Annex) with a timestamp equal to the current value of time, in case motion information calculated from the motion sensor is contradicted by motion information calculated from the internal GNSS receiver or from the external GNSS facility. For the purpose of detecting such contradictions, the median value of the speed differences between these sources shall be used, as specified below:

- every 10 seconds maximum, the absolute value of the difference between the vehicle speed estimated from the GNSS and the one estimated from the motion sensor shall be computed.
- all the computed values in a time window containing the last five minutes of movement shall be used to compute the median value.
- the median value shall be computed as the average of 80% of the remaining values, after having eliminated the highest ones in absolute values

The Vehicle Motion Conflict event shall be triggered if the median value is above 10 Km/h for five uninterrupted minutes of vehicle movement. Other independent sources of vehicle motion detection may optionnally be used, so that a more reliable detection of tachograph manipulations is provided. (Note: the use of the median on the last 5 minutes is applied to mitigate the risk of measurement outliers and transient values). This event shall not be triggered in the following conditions: a) during a ferry/train crossing, b) when the position information from the GNSS receiver shall not be available and c) while in calibration mode.

# Appendix 13 - ITS Interface

## TABLE OF CONTENTS

<b>1. INTRODUCTION</b>	<b>513</b>
<b>2. SCOPE</b>	<b>513</b>
<b>2.1. Acronyms, definitions and notations</b>	<b>514</b>
<b>3. REFERENCED REGULATIONS AND STANDARDS</b>	<b>514</b>
<b>4. INTERFACE WORKING PRINCIPLES</b>	<b>515</b>
<b>4.1. Preconditions to data transfer via the ITS interface</b>	<b>515</b>
4.1.1 .. Data provided through the ITS interface	515
4.1.2 .. Content of the Data	515
4.1.3 .. ITS Applications	515
<b>4.2. Communication technology</b>	<b>515</b>
<b>4.3. PIN authorization</b>	<b>517</b>
<b>4.4. Message Format</b>	<b>518</b>
<b>4.5. Driver consent</b>	<b>524</b>
<b>4.6. Standard data retrieval</b>	<b>524</b>
<b>4.7. Personal data retrieval</b>	<b>524</b>
<b>4.8. Event and fault data retrieval</b>	<b>525</b>



## 1. Introduction

This Appendix specifies the design and the procedures to follow in order to implement the interface with Intelligent Transport Systems (ITS) as required in Article 10 of Regulation (EU) N°. 165/2014 (*the Regulation*).

*The Regulation* specifies that the tachographs of vehicles may be equipped with standardised interfaces allowing the data recorded or produced by tachograph to be used in operational mode, by an external device, provided that the following conditions are met:

- (a) the interface does not affect the authenticity and the integrity of the data of the tachograph;
- (b) the interface complies with the detailed provisions of Article 11 of the Regulation;
- (c) the external device connected to the interface has access to personal data, including geopositioning data, only after the verifiable consent of the driver to whom the data relates.

## 2. Scope

The scope of this Appendix is to specify how applications hosted on external devices can via a Bluetooth® connection obtain data (*the Data*) from a tachograph.

*The Data* available via this interface is described in the Annex 1 of the present document. This interface does not prohibit the implementation of other interfaces (e.g. via the CAN bus) to transmit the data of the VU to other vehicle processing units.

This Appendix specifies:

- *The Data* available through the ITS interface
- The Bluetooth® profile that is used to transfer the data
- The enquiry and download procedures and sequence of operations
- The pairing mechanism between the tachograph and the external device
- The consent mechanism available to the driver

For clarification, this Annex does not specify:

- The collection of *the Data* operation and management within the VU (which shall be specified elsewhere within *the Regulation* or otherwise shall be a function of product design).
- The form of presentation of collected data to application hosted on the external device.

- Data security provisions above what provides Bluetooth® (such as encryption) concerning the content of *the Data* (which shall be specified elsewhere within *the Regulation* [Appendix 10 Common Security Mechanisms]).
- The Bluetooth® protocols used by the ITS interface

### 2.1. Acronyms, definitions and notations

The following acronyms and definitions specific to this Appendix are used in this appendix:

<b><i>the Communication</i></b>	exchange of information/data between a master unit (i.e. the tachographs) and an external unit through the ITS interface over Bluetooth®.
<b><i>the Data</i></b>	Data sets as specified in Annex 1.
<b><i>the Regulation</i></b>	Regulation (EU) N°. 165/2014 of the European Parliament and of the Council of 4 February 2014 on tachographs in road transport, repealing Council Regulation (EEC) N°. 3821/85 on recording equipment in road transport and amending Regulation (EC) N°. 561/2006 of the European Parliament and of the Council on the harmonisation of certain social legislation relating to road transport

<b>BR</b>	Basic Rate
<b>EDR</b>	Enhanced Data Rate
<b>GNSS</b>	Global Navigation Satellite System
<b>IRK</b>	Identity Resolution Key
<b>ITS</b>	Intelligent Transport System
<b>LE</b>	Low Energy
<b>PIN</b>	Personal Identification Number
<b>PUC</b>	Personal Unblocking Code
<b>SID</b>	Service Identifier
<b>SPP</b>	Serial Port Profile
<b>SSP</b>	Secure Simple Pairing
<b>TRTP</b>	Transfer Request Parameter
<b>TREP</b>	Transfer Response Parameter
<b>VU</b>	Vehicle Unit

### 3. Referenced Regulations and Standards

The specification defined in this Appendix refers to and depends upon all or parts of the following regulations and standards. Within the clauses of this Appendix the relevant standards, or relevant clauses of standards, are specified. In the event of any contradiction the clauses of this Appendix shall take precedence.

Regulations and standards referenced in this Appendix are:

- Regulation (EU) N°. 165/2014 of the European Parliament and of the Council of 4 February 2014 on tachographs in road transport, repealing Council Regulation (EEC) N°. 3821/85 on recording equipment in road transport and amending Regulation (EC) N°. 561/2006 of the European Parliament and of the Council on the harmonisation of certain social legislation relating to road transport.
- Regulation (EC) N°. 561/2006 of the European Parliament and of the Council of 15 March 2006 on the harmonisation of certain social legislation relating to road transport and amending Council Regulations (EEC) N°. 3821/85 and (EC) No 2135/98 and repealing Council Regulation (EEC) N°. 3820/85.
- ISO 16844 – 4 : Road vehicles – Tachograph systems – Part 4: Can interface
- ISO 16844 – 7 : Road vehicles – Tachograph systems – Part 7: Parameters
- Bluetooth® – Serial Port Profile – V1.2
- Bluetooth® – Core Version 4.2
- NMEA 0183 V4.1 protocol

#### 4. Interface working principles

##### 4.1. Preconditions to data transfer via the ITS interface

The VU shall be responsible to keep updated and maintain the data to be stored in the VU, without any involvement of the ITS interface. The means by which this is achieved is internal to the VU, specified elsewhere in the Regulation, and is not specified in this Appendix.

##### 4.1.1 Data provided through the ITS interface

The VU shall be responsible to update the data that will be available through the ITS interface at a frequency determined within VU procedures, without any involvement of ITS interface. The VU data shall be used as a basis to populate and update *the Data*, the means by which this is achieved is specified elsewhere in *the Regulation* or if there is no such specification is a function of product design and is not specified in this Appendix.

##### 4.1.2 Content of the Data

The content of *the Data* shall be as specified in Annex 1 of this appendix.

##### 4.1.3 ITS Applications

ITS applications will be using the data made available through the ITS interface for instance to optimize driver activities management while respecting the Regulation, to detect possible faults of the tachograph or to use the GNSS data. The specification of the applications is not within the scope of this Appendix.

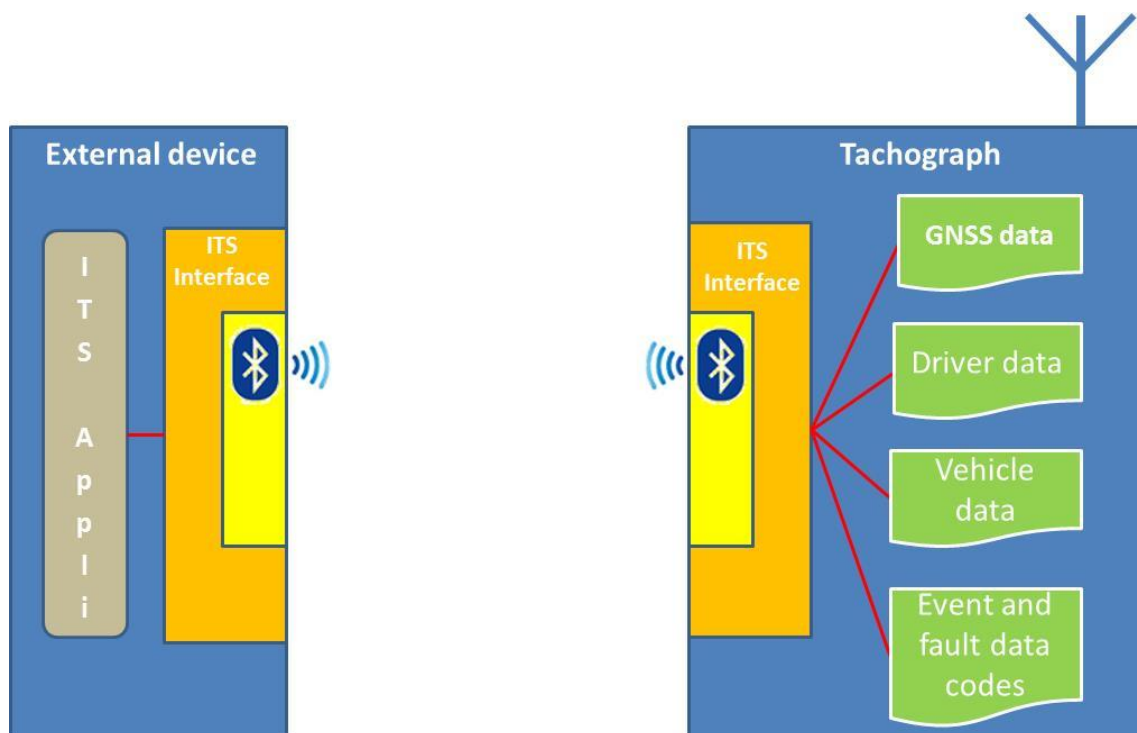
#### 4.2. Communication technology

The Data exchange using the ITS interface shall be performed via a Bluetooth® interface compatible via version 4.2 or later. Bluetooth® operates in the unlicensed industrial, scientific and medical (ISM) band at 2.4 to 2.485 GHz. Bluetooth® 4.2 offers enhanced privacy and security mechanisms and increases speed and reliability of data transfers. For the purpose of this specification is Bluetooth® class 2 radio used with a range up to 10 meters. More information on Bluetooth® 4.2 is available on [www.bluetooth.com](http://www.bluetooth.com) ([https://www.bluetooth.org/en-us/specification/adopted-specifications?\\_ga=1.215147412.2083380574.1435305676](https://www.bluetooth.org/en-us/specification/adopted-specifications?_ga=1.215147412.2083380574.1435305676)).

The Communication shall be established with the communications equipment after a pairing process has been completed by an authorized device. As Bluetooth® is using a master/slave model to control when and where devices can send data, the tachograph will play the role of master while the external device will be the slave.

When an external device comes within range of the VU for the first time, the Bluetooth® pairing process can be initiated (see also annex 2). The devices share their addresses, names, and profiles and common secret key, which allows them to bond whenever they're together in the future. Once this step is completed, the external device is trusted and is in state to initiate requests to download data from the tachograph. It is not foreseen to add encryption mechanisms beyond what Bluetooth® provides. However, if additional security mechanisms are needed, this will be done in accordance with Appendix 10 Common Security Mechanisms.

The overall communication principle is described in the following figure.



The SPP (Serial Port Profile) profile of Bluetooth® shall be used to transfer data from the VU to the external device.

#### 4.3. PIN authorization

For security reasons, the VU shall enforce a PIN code authorization system separated from the Bluetooth pairing. Each VU shall be able to generate PIN codes for authentication purposes composed of at least 4 digits. Every time an external device pairs with the VU, it must provide the correct PIN code before receiving any data.

Succeeding entering the PIN shall result in putting the device on the whitelist. The whitelist shall store at least 64 devices paired with the particular VU.

Failing to provide the correct PIN code three times in a row shall result in putting temporarily the device on the blacklist. While blacklisted, every new attempt from the device shall be rejected. Further failure to provide the correct PIN code three times in a row shall result in increasingly longer ban duration (See table 1). Providing the correct PIN code shall reset the ban duration and the number of attempt. Figure 1 in Annex 2 represents the sequence diagram of a PIN validation attempt.

<i>Number of consecutive failure</i>	<i>Ban duration</i>
3	30 seconds
6	5 minutes
9	1 hour
12	24 hours
15	Permanent

**Table 50: Ban duration depending on the number of consecutive failure to provide the correct PIN code**

Failing to provide the correct PIN code fifteen times (5x3) in a row shall result in a permanent blacklisting of the ITS Unit. Only providing the correct PUC code shall overturn this permanent ban.

The PUC code shall be composed of 8 digits and provided by the manufacturer with the VU. Failing to provide the correct PUC code ten times in a row will irrevocably blacklist the ITS Unit.

While the manufacturer may offers an option to change the PIN code directly through the VU, the PUC code shall not be alterable. Modifying the PIN code, if possible, shall require to enter the current PIN code directly in the VU.

Furthermore any devices stored in the whitelist shall be kept until manual removal of by the user (e.g. via the man-machine-interface of the VU or other means). By doing so lost or stolen ITS-units may be removed from the whitelist. Also, any ITS Unit leaving the Bluetooth connection range for more than 24 hours shall be automatically removed from the VU whitelist and must provide the correct PIN code again when the connection is established again.

The format of the messages between the VU interface and the VU are not provided but left to the discretion of the manufacturer. Said manufacturer shall however ensure the message format between the ITS Unit and the VU interface is respected (see ASN.1 specifications).

Any data request shall thus be met with the proper verification of the sender’s credential before any form of treatment. Figure 2 of Annex 2 represents the sequence diagram for this procedure. Any blacklisted device shall receive an automatic rejection, any non-blacklisted non-whitelisted device shall receive a PIN request it needs to fulfill before resending its data request.

#### 4.4. Message Format

All messages exchanged between the ITS Unit and the VU interface shall be formatted with a structure consisting of three parts: A header composed by a target byte (TGT), a source byte (SRC) and a length byte (LEN).

The data field composed by a service identifier byte (SID) and a variable amount of data bytes (maximum 255).

The checksum byte is the 1 byte sum series modulo 256 of all the bytes of the message excluding the CS itself.

The message shall be Big Endian.

<i>Header</i>			<i>Data Field</i>					<i>Checksum</i>
TGT	SRC	LEN	SID	TRTP	CC	CM	DATA	CS
3 bytes			Max. 255 bytes					1 byte

**Table 51: General message format.**

#### *Header*

TGT and SRC : the ID of the Target (TGT) and Source (SRC) devices of the message. The VU Interface shall have the default ID “EE”. This ID cannot be changed. The ITS Unit shall use the default ID “A0” for its first message of the communication session. The VU Interface shall then assign an unique ID to the ITS Unit and informs it of this ID for future messages during the session.

The LEN byte shall only take into account the “DATA” part of the Data Field (see Table 2), the 4 first bytes are implicit.

The VU Interface shall confirm the authenticity of the message’s sender by cross-checking its own IDList with the Bluetooth data by checking the ITS Unit listed at the provided ID is currently in the range of the Bluetooth connection.

*Data Field*

Besides the SID, the Data Field shall also contain other parameters : a transfer request parameter (TRTP) and Counter bytes.

If the data which need to be carried is too long than the available space in one message, it will be split in several submessage. Each submessage shall have the same Header and SID, but will contain a 2-bytes counter, Counter Current (CC) and Counter Max (CM), to indicate the submessage number. To enable error checking and abort the receiving device acknowledges every submessage. The receiving device can accept the submessage, ask for it to be re-transmitted, request the sending device to start again or abort the transmission.

If not used, CC and CM shall be given the value 0xFF.

For instance, the following message

<i>HEADER</i>	<i>SID</i>	<i>TRTP</i>	<i>CC</i>	<i>CM</i>	<i>DATA</i>	<i>CS</i>
3 bytes	Longer than 255 bytes					1 byte

Shall be transmitted as such:

<i>HEADER</i>	<i>SID</i>	<i>TRTP</i>	<i>01</i>	<i>n</i>	<i>DATA</i>	<i>CS</i>
3 bytes	255 bytes					1 byte

<i>HEADER</i>	<i>SID</i>	<i>TRTP</i>	<i>02</i>	<i>n</i>	<i>DATA</i>	<i>CS</i>
3 bytes	255 bytes					1 byte

...

<i>HEADER</i>	<i>SID</i>	<i>TRTP</i>	<i>N</i>	<i>N</i>	<i>DATA</i>	<i>CS</i>
3 bytes	Max. 255 bytes					1 byte

Table 3 contains the messages the VU and the ITS Unit shall be able to exchange. The content of each parameter is given in hexadecimal. Aren't represented in the table CC and CM for clarity, see above for complete format.

<i>Message</i>	<i>Header</i>			<i>DATA</i>			<i>Checksum</i>
	TGT	SRC	LEN	SID	TRTP	DATA	
<i>RequestPIN</i>	<i>ITSID</i>	EE	00	01	FF		
<i>SendITSID</i>	<i>ITSID</i>	EE	01	02	FF	<i>ITSID</i>	
<i>SendPIN</i>	EE	<i>ITSID</i>	04	03	FF	4*INTEGER (0..9)	
<i>PairingResult</i>	<i>ITSID</i>	EE	01	04	FF	BOOLEAN (T/F)	
<i>SendPUC</i>	EE	<i>ITSID</i>	08	05	FF	8*INTEGER (0..9)	
<i>BanLiftingResult</i>	<i>ITSID</i>	EE	01	06	FF	BOOLEAN (T/F)	
<i>RequestRejected</i>	<i>ITSID</i>	EE	08	07	FF	Time	
<i>RequestData</i>							
standardTachData	EE	<i>ITSID</i>	01	08	01		
personalTachData	EE	<i>ITSID</i>	01	08	02		
gnssData	EE	<i>ITSID</i>	01	08	03		
standardEventData	EE	<i>ITSID</i>	01	08	04		
personalEventData	EE	<i>ITSID</i>	01	08	05		
standardFaultData	EE	<i>ITSID</i>	01	08	06		
manufacturerData	EE	<i>ITSID</i>	01	08	07		
<i>RequestAccepted</i>	<i>ITSID</i>	EE	Len	09	TREP	Data	
<i>DataUnavailable</i>							

---



<i>Message</i>	<i>Header</i>		<i>DATA</i>				<i>Checksum</i>
	TGT	SRC	LEN	SID	TRTP	DATA	
No data available	<i>ITSID</i>	EE	02	0A	TREP	10	
Personal data not shared	<i>ITSID</i>	EE	02	0A	TREP	11	
<i>NegativeAnswer</i>							
General reject	<i>ITSID</i>	EE	02	0B	SID Req	10	
Service not supported	<i>ITSID</i>	EE	02	0B	SID Req	11	
Sub function not supported	<i>ITSID</i>	EE	02	0B	SID Req	12	
Incorrect message length	<i>ITSID</i>	EE	02	0B	SID Req	13	
Conditions not correct or request sequence error	<i>ITSID</i>	EE	02	0B	SID Req	22	
Request out of range	<i>ITSID</i>	EE	02	0B	SID Req	31	
Response pending	<i>ITSID</i>	EE	02	0B	SID Req	78	
ITSID Mismatch	<i>ITSID</i>	EE	02	0B	SID Req	FC	
ITSID Not Found	<i>ITSID</i>	EE	02	0B	SID Req	FB	

Table 52: Detailed message content.

*RequestPIN (SID 01)*

This message is issued by the VU Interface if a non-blacklisted but non-whitelisted ITS unit is sending any data request.

*SendITSID (SID 02)*

This message is issued by the VU Interface whenever a new device is sending a request. This device shall use the default ID "A0" before getting assigned an unique ID for the communication session.

*SendPIN (SID 03)*

This message is issued by the ITS Unit to be whitelisted from the VU interface. The content of this message is a 4 INTEGER between 0 and 9 code.

*PairingResult (SID 04)*

This message is issued by the VU Interface to inform the ITS Unit if the PIN code it sent was correct. The content of this message shall be a BOOLEAN with the value “True” if the PIN code was correct and “False” otherwise.

*SendPUC (SID 05)*

This message is issued by the ITS Unit to lift a blacklist sanction from the VU interface. The content of this message is a 8 INTEGER between 0 and 9 code.

*BanLiftingResult (SID 06)*

This message is issued by the VU Interface to inform the ITS Unit if the PUC code it sent was correct. The content of this message shall be a BOOLEAN with the value “True” if the PUC code was correct and “False” otherwise.

*RequestRejected (SID 07)*

This message is issued by the VU Interface as a reply to any message from a blacklisted ITS Unit except “SendPUC”. The message shall contain the remaining time the ITS Unit is blacklisted, following the “Time” sequence format as defined in Annex 3.

*RequestData (SID 08)*

This message for data accessing is issued by the ITS Unit. A one byte transfer request parameter (TRTP) indicates the type of data required. There are several types of data:

- standardTachData (TRTP 01): Data available from the tachograph classified as non-personal.
- personalTachData (TRTP 02): Data available from the tachograph classified as personal.
- gnssData (TRTP 03): GNSS data, always personal.
- standardEventData (TRTP 04): Recorded event data classified as non-personal.
- personalEventData (TRTP 05): Recorded event data classified as personal.
- standardFaultData (TRTP 06): Recorded faults classified as non-personal.
- manufacturerData (TRTP 07): data made available by the manufacturer.

See Annex 3 of this appendix for more information about the content of each data type.

See Appendix 12 for more information about the format and content of GNSS data.

See Annex IB and IC for more information about event data code and faults.

#### *ResquestAccepted* (SID 09)

This message is issued by the VU Interface if a ITS Unit “RequestData” message has been accepted. This message contains a 1-byte TREP, which is the TRTP byte of the associated RequestData message, and all the data of the requested type.

#### *DataUnavailable* (SID 0A)

This message is issued by the VU Interface if, for a certain reason, the requested data aren’t available to be sent to a whitelisted ITS Unit. The message contains a 1byte TREP which is the TRTP of the required data and a 1 byte error code specified in the table 3. The Following codes are available:

- No data available (10) : The VU interface can’t access the VU data for unspecified reasons.
- Personal data not shared (11) : The ITS Unit tries to retrieve personal data when they are not shared.

#### *NegativeAnswer* (SID 0B)

These messages are issued by the VU Interface if a request cannot be completed for any other reason than the unavailability of the data. These messages are typically the result of a bad request format (Length, SID, ITSID...) but aren’t limited to that. The TRTP in the Data Field contains the SID of the request. The Data Field contains a code identifying the reason of the negative answer. The following codes are available:

- General Reject (code : 10)  
The action can’t be performed for a reason which isn’t cited below nor in section (Enter *DataUnavailable* section number).
- Service not supported (code : 11)  
The request’s SID isn’t understood.
- Sub function not supported (code : 12)  
The request’s TRTP isn’t understood. It can be for instance missing or out of accepted values.
- Incorrect message length (code : 13)  
The length of the received message is wrong (mismatch between the LEN byte and the actual message length).
- Conditions not correct or request sequence error (code : 22)  
The required service is not active or the sequence of request messages is not correct
- Request out of range (code : 33)  
The request parameter record (data field) is not valid
- Response pending (code : 78)  
The action requested cannot be completed in time and the VU is not ready to accept another request.
- *ITSID* Mismatch (code : FB)

The SRC *ITSID* doesn't match the associated device after comparison with the Bluetooth information.

- *ITSID* Not Found (code : FC)

The SRC *ITSID* isn't associated with any device.

Lines 1 through 72 (**FormatMessageModule**) of the ASN.1 code in Annex 3 specify the messages format as described in table 3. More details about the messages content is given below.

#### 4.5. Driver consent

All the data available are classified as either standard or personal. Personal data shall only be accessible if the driver gave his/her consent, accepting his/her tachograph personal data can leave the vehicle network for third party applications.

Driver consent is given when, at first insertion of a given driver card or workshop card currently unknown to the vehicle unit, the cardholder is invited to express his consent for tachograph related personal data output through the optional ITS interface. (see also Annex I C paragraph 3.6.2).

The consent status (enabled/disabled) is recorded in the memory of the tachograph.

In case of multiple drivers, only the personal data about the drivers who gave their consent shall be shared with the ITS interface. For instance, if there's two drivers in the vehicle, and only the first driver accepted to share his personal data, the ones concerning the second driver shall not be shared.

#### 4.6. Standard data retrieval

Figure 3 of Annex 2 represents the sequence diagrams of a valid request sent by the ITS Unit to access standard data. The ITS Unit is properly whitelisted and isn't requesting personal data, no further verification is required. The diagrams consider the proper procedure illustrated in Figure 2 of Annex 2 has already been followed. They can be equated to the *REQUEST TREATMENT* gray box of Figure 2.

Amongst available data, shall be considered standard:

- standardTachData (TRTP 01)
- StandardEventData (TRTP 04)
- standardFaultData (TRTP 06)

#### 4.7. Personal data retrieval

Figure 4 of Annex 2 represents the sequence diagram for personal data request processing. As previously stated, the VU interface shall only send personal data if the driver has given his explicit consent (see also 4.5). Otherwise, the request must be automatically rejected.

Amongst available data, shall be considered personal:

- personalTachData (TRTP 02)
- gnssData (TRTP 03)
- personalEventData (TRTP 05)
- manufacturerData (TRTP 07)

#### 4.8. Event and fault data retrieval

ITS units shall be able to request events data containing the list of all the unexpected events. These data are considered standard or personal, see Annex 3. The content of each event is in accordance with the documentation provided in Annex 1 of this appendix.

# ANNEX 1

## LIST OF AVAILABLE DATA THROUGH THE ITS INTERFACE

Data	Source	Data classification (personal/not personal)
VehicleIdentificationNumber	Vehicle Unit	not personal
CalibrationDate	Vehicle Unit	not personal
TachographVehicleSpeed speed instant t	Vehicle Unit	personal
Driver1WorkingState Selector driver	Vehicle Unit	personal
Driver2WorkingState	Vehicle Unit	personal
DriveRecognize Speed Threshold detected	Vehicle Unit	not personal
Driver1TimeRelatedStates Weekly day time	Driver Card	personal
Driver2TimeRelatedStates	Driver Card	personal
DriverCardDriver1	Vehicle Unit	not personal
DriverCardDriver2	Vehicle unit	not personal
OverSpeed	Vehicle Unit	personal
TimeDate	Vehicle Unit	not personal
HighResolutionTotalVehicleDistance	Vehicle Unit	not personal
ServiceComponentIdentification	Vehicle Unit	not personal
ServiceDelayCalendarTimeBased	Vehicle Unit	not personal
Driver1Identification	Driver Card	personal
Driver2Identification	Driver Card	personal
NextCalibrationDate	Vehicle Unit	not personal
Driver1ContinuousDrivingTime	Driver Card	personal
Driver2ContinuousDrivingTime	Driver Card	personal
Driver1CumulativeBreakTime	Driver Card	personal
Driver2CumulativeBreakTime	Driver Card	personal
Driver1CurrentDurationOfSelectedActivity	Driver Card	personal
Driver2CurrentDurationOfSelectedActivity	Driver Card	personal
SpeedAuthorised	Vehicle Unit	not personal
TachographCardSlot1	Driver Card	not personal
TachographCardSlot2	Driver Card	not personal
Driver1Name	Driver Card	personal
Driver2Name	Driver Card	personal
OutOfScopeCondition	Vehicle Unit	not personal
ModeOfOperation	Vehicle Unit	not personal
Driver1CumulatedDrivingTimePreviousAndCurrentWeek	Driver Card	personal
Driver2CumulatedDrivingTimePreviousAndCurrentWeek	Driver Card	personal
EngineSpeed	Vehicle Unit	personal
RegisteringMemberState	Vehicle Unit	not personal
VehicleRegistrationNumber	Vehicle Unit	not personal
Driver1EndOfLastDailyRestPeriod	Driver Card	personal
Driver2EndOfLastDailyRestPeriod	Driver Card	personal
Driver1EndOfLastWeeklyRestPeriod	Driver Card	personal
Driver2EndOfLastWeeklyRestPeriod	Driver Card	personal
Driver1EndOfSecondLastWeeklyRestPeriod	Driver Card	personal
Driver2EndOfSecondLastWeeklyRestPeriod	Driver Card	personal
Driver1CurrentDailyDrivingTime	Driver Card	personal
Driver2CurrentDailyDrivingTime	Driver Card	personal
Driver1CurrentWeeklyDrivingTime	Driver Card	personal
Driver2CurrentWeeklyDrivingTime	Driver Card	personal
Driver1TimeLeftUntilNewDailyRestPeriod	Driver Card	personal
Driver2TimeLeftUntilNewDailyRestPeriod	Driver Card	personal
Driver1CardExpiryDate	Driver Card	personal
Driver2CardExpiryDate	Driver Card	personal
Driver1CardNextMandatoryDownloadDate	Driver Card	personal
Driver2CardNextMandatoryDownloadDate	Driver Card	personal
TachographNextMandatoryDownloadDate	Vehicle Unit	not personal
Driver1TimeLeftUntilNewWeeklyRestPeriod	Driver Card	personal
Driver2TimeLeftUntilNewWeeklyRestPeriod	Driver Card	personal
Driver1NumberOfTimes9hDailyDrivingTimesExceeded	Driver Card	personal
Driver2NumberOfTimes9hDailyDrivingTimesExceeded	Driver Card	personal
Driver1CumulativeUninterruptedRestTime	Driver Card	personal
Driver2CumulativeUninterruptedRestTime	Driver Card	personal
Driver1MinimumDailyRest	Driver Card	personal
Driver2MinimumDailyRest	Driver Card	personal
Driver1MinimumWeeklyRest	Driver Card	personal
Driver2MinimumWeeklyRest	Driver Card	personal
Driver1MaximumDailyPeriod	Driver Card	personal
Driver2MaximumDailyPeriod	Driver Card	personal
Driver1MaximumDailyDrivingTime	Driver Card	personal
Driver2MaximumDailyDrivingTime	Driver Card	personal
Driver1NumberOfUsedReducedDailyRestPeriods	Driver Card	personal
Driver2NumberOfUsedReducedDailyRestPeriods	Driver Card	personal
Driver1RemainingCurrentDrivingTime	Driver Card	personal
Driver2RemainingCurrentDrivingTime	Driver Card	personal
GnssPosition	Vehicle Unit	personal

**2°) CONTINUOUS GNSS DATA AVAILABLE AFTER DRIVER CONSENT**

See Appendix 12 – GNSS.

**3°) EVENT CODES AVAILABLE WITHOUT DRIVER CONSENT**

<i>Event</i>	<i>Storage rules</i>	<i>Data to be recorded per event</i>
Insertion of a non-valid card	- the 10 most recent events.	- date and time of event, - card(s) type, number, issuing Member State and generation of the card creating the event. - number of similar events that day
Card conflict	- the 10 most recent events.	- date and time of beginning of event, - date and time of end of event, - card(s) type, number, issuing Member State and generation of the two cards creating the conflict.
Last card session not correctly closed	- the 10 most recent events.	- date and time of card insertion, - card(s) type, number, issuing Member State and generation, - last session data as read from the card: - date and time of card insertion, - VRN, Member State of registration and VU generation.



<i>Event</i>	<i>Storage rules</i>	<i>Data to be recorded per event</i>
Power supply interruption (2)	<ul style="list-style-type: none"> <li>- the longest event for each of the 10 last days of occurrence,</li> <li>- the 5 longest events over the last 365 days.</li> </ul>	<ul style="list-style-type: none"> <li>- date and time of beginning of event,</li> <li>- date and time of end of event,</li> <li>- card(s) type, number, issuing Member State and generation of any card inserted at beginning and/or end of the event,</li> <li>- number of similar events that day.</li> </ul>
Communication error with the remote communication facility	<ul style="list-style-type: none"> <li>- the longest event for each of the 10 last days of occurrence,</li> <li>- the 5 longest events over the last 365 days.</li> </ul>	<ul style="list-style-type: none"> <li>- date and time of beginning of event,</li> <li>- date and time of end of event,</li> <li>- card(s) type, number, issuing Member State and generation of any card inserted at beginning and/or end of the event,</li> <li>- number of similar events that day.</li> </ul>
Absence of position information from GNSS receiver	<ul style="list-style-type: none"> <li>- the longest event for each of the 10 last days of occurrence,</li> <li>- the 5 longest events over the last 365 days.</li> </ul>	<ul style="list-style-type: none"> <li>- date and time of beginning of event,</li> <li>- date and time of end of event,</li> <li>- card(s) type, number, issuing Member State and generation of any card inserted at beginning and/or end of the event,</li> <li>- number of similar events that day.</li> </ul>

<i>Event</i>	<i>Storage rules</i>	<i>Data to be recorded per event</i>
Motion data error	<ul style="list-style-type: none"> <li>- the longest event for each of the 10 last days of occurrence,</li> <li>- the 5 longest events over the last 365 days.</li> </ul>	<ul style="list-style-type: none"> <li>- date and time of beginning of event,</li> <li>- date and time of end of event,</li> <li>- card(s) type, number, issuing Member State and generation of any card inserted at beginning and/or end of the event,</li> <li>- number of similar events that day.</li> </ul>
Vehicle motion conflict	<ul style="list-style-type: none"> <li>- the longest event for each of the 10 last days of occurrence,</li> <li>- the 5 longest events over the last 365 days.</li> </ul>	<ul style="list-style-type: none"> <li>- date and time of beginning of event,</li> <li>- date and time of end of event,</li> <li>- card(s) type, number, issuing Member State and generation of any card inserted at beginning and/or end of the event,</li> <li>- number of similar events that day.</li> </ul>
Security breach attempt	<ul style="list-style-type: none"> <li>- the 10 most recent events per type of event.</li> </ul>	<ul style="list-style-type: none"> <li>- date and time of beginning of event,</li> <li>- date and time of end of event (if relevant),</li> <li>- card(s) type, number, issuing Member State and generation of any card inserted at beginning and/or end of the event,</li> <li>- type of event.</li> </ul>
Time conflict	<ul style="list-style-type: none"> <li>- the longest event for each of the 10 last days of occurrence,</li> <li>- the 5 longest events over the last 365 days.</li> </ul>	<ul style="list-style-type: none"> <li>- recording equipment date and time</li> <li>- GNSS date and time,</li> <li>- card(s) type, number, issuing Member State and generation of any card inserted at beginning and/or end of the event,</li> <li>- number of similar events that day.</li> </ul>

**4°) EVENT CODES AVAILABLE WITH DRIVER CONSENT**

<i>Event</i>	<i>Storage rules</i>	<i>Data to be recorded per event</i>
Driving without an appropriate card	<ul style="list-style-type: none"> <li>- the longest event for each of the 10 last days of occurrence,</li> <li>- the 5 longest events over the last 365 days.</li> </ul>	<ul style="list-style-type: none"> <li>- date and time of beginning of event,</li> <li>- date and time of end of event,</li> <li>- card(s) type, number, issuing Member State and generation of any card inserted at beginning and/or end of the event,</li> <li>- number of similar events that day.</li> </ul>
Card insertion while driving	<ul style="list-style-type: none"> <li>- the last event for each of the 10 last days of occurrence,</li> </ul>	<ul style="list-style-type: none"> <li>- date and time of the event,</li> <li>- card(s) type, number, issuing Member State and generation,</li> <li>- number of similar events that day</li> </ul>
Over speeding (1)	<ul style="list-style-type: none"> <li>- the most serious event for each of the 10 last days of occurrence (i.e. the one with the highest average speed),</li> <li>- the 5 most serious events over the last 365 days.</li> <li>- the first event having occurred after the last calibration</li> </ul>	<ul style="list-style-type: none"> <li>- date and time of beginning of event,</li> <li>- date and time of end of event,</li> <li>- maximum speed measured during the event,</li> <li>- arithmetic average speed measured during the event,</li> <li>- card type, number, issuing Member State and generation of the driver card (if applicable),</li> <li>- number of similar events that day.</li> </ul>

**5°) FAULT DATA CODES AVAILABLE WITHOUT DRIVER CONSENT**

<i>Fault</i>	<i>Storage rules</i>	<i>Data to be recorded per fault</i>
Card fault	- the 10 most recent driver card faults.	- date and time of beginning of fault, - date and time of end of fault, - card(s) type, number, issuing Member State and generation.
Recording equipment faults	- the 10 most recent faults for each type of fault, - the first fault after the last calibration.	- date and time of beginning of fault, - date and time of end of fault, - type of fault, - card(s) type, number and issuing Member State and generation of any card inserted at beginning and/or end of the fault.

This fault shall be triggered for any of these failures, while not in calibration mode:

- VU internal fault
- Printer fault
- Display fault
- Downloading fault
- Sensor fault
- GNSS receiver or external GNSS facility fault
- Remote Communication facility fault

**6°) MANUFACTURER SPECIFIC EVENTS AND FAULTS WITHOUT DRIVER CONSENT**

<i>Event or Fault</i>	<i>Storage rules</i>	<i>Data to be recorded per event</i>
To be defined by Manufacturer	To be defined by Manufacturer	To be defined by Manufacturer



# ANNEX 2

SEQUENCE DIAGRAMS OF MESSAGES EXCHANGES WITH THE ITS UNIT.

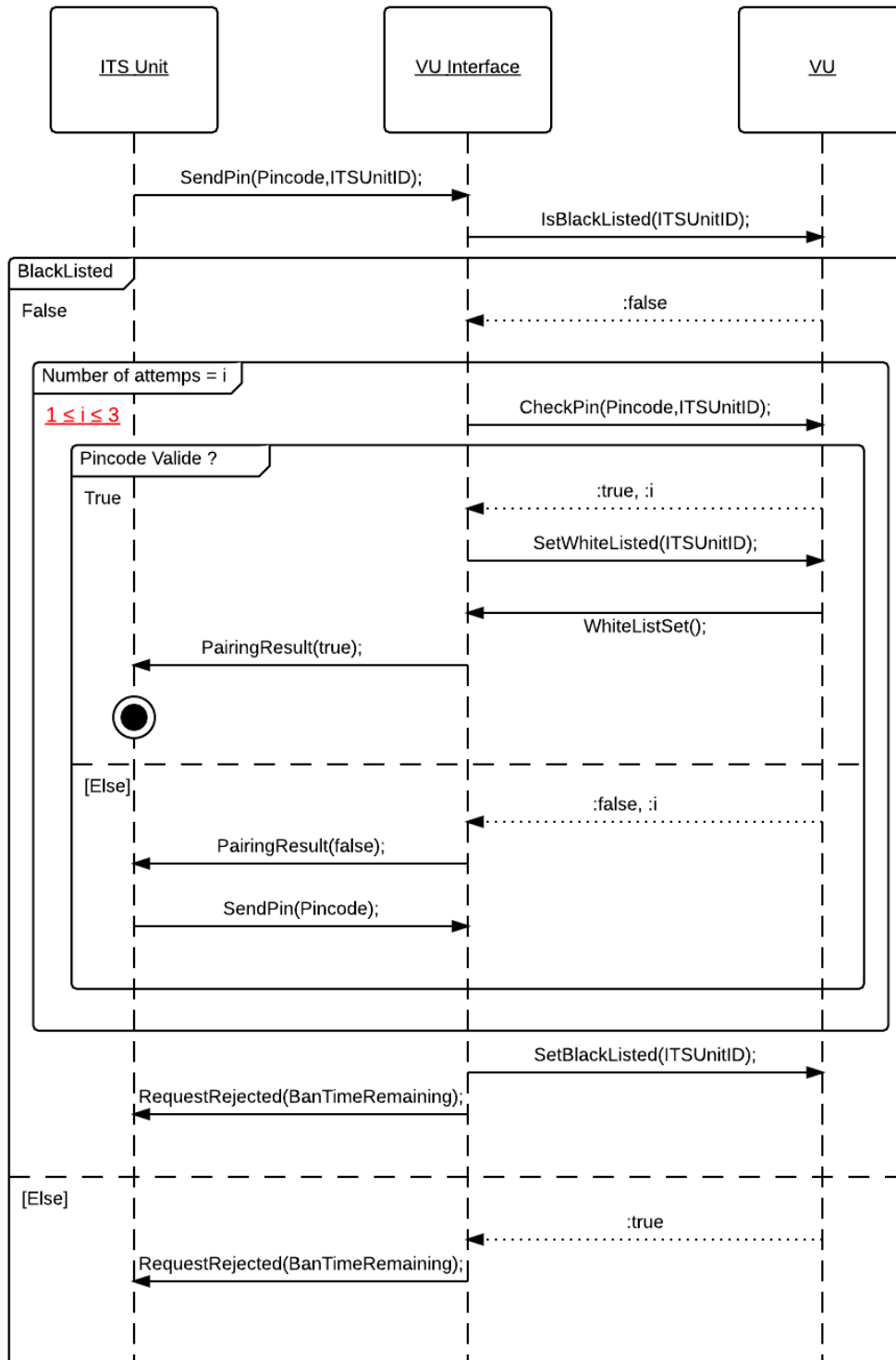


Figure 18. Sequence Diagram for PIN validation attempt

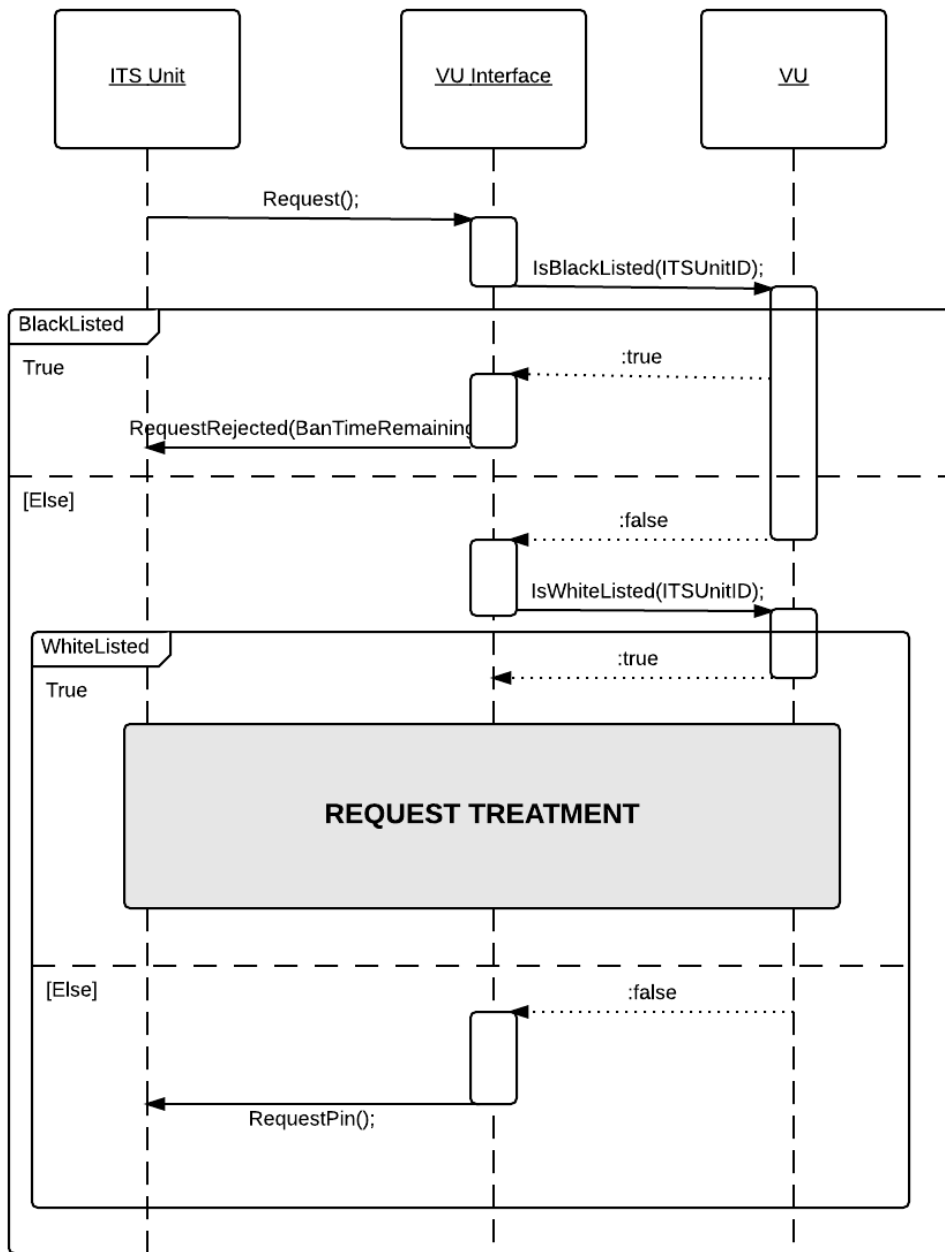


Figure 19. Sequence Diagram for ITS Unit's authorization verification



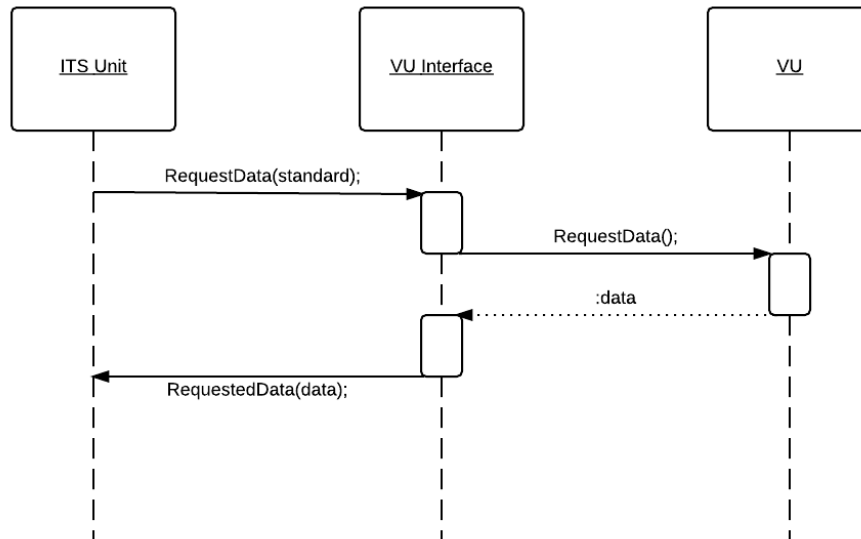


Figure 20. Sequence Diagram to process a request for data classified as non-personal (after correct PIN access)

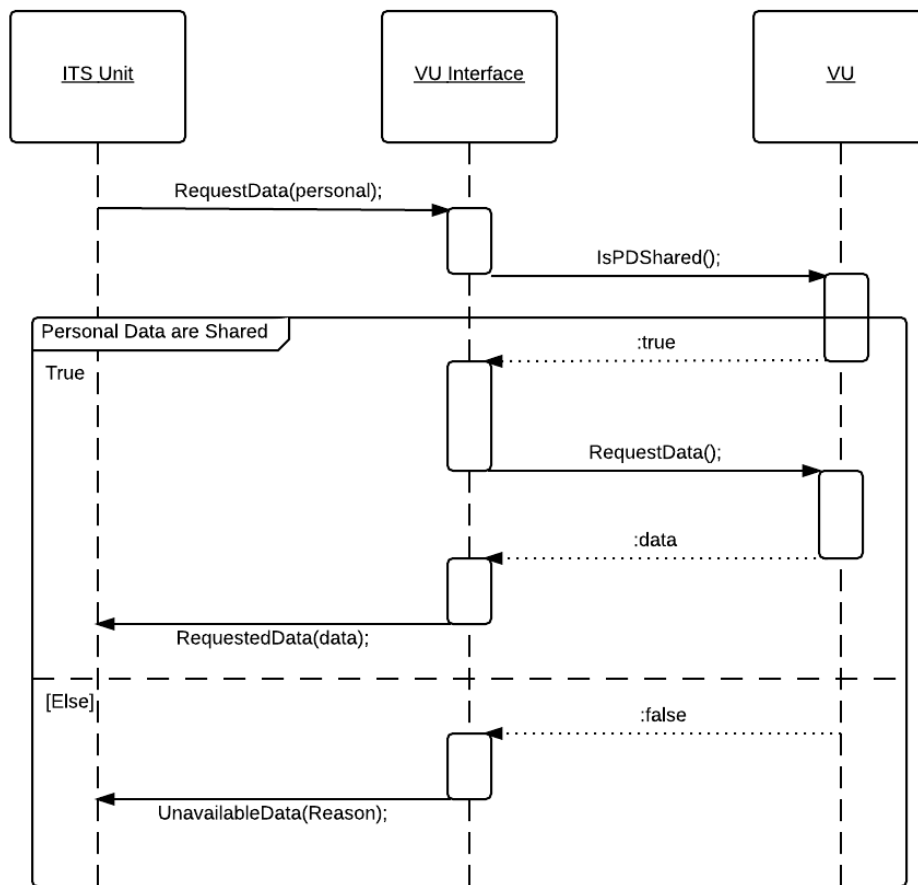


Figure 21. Sequence Diagram to process a request for data classified as personal (after correct PIN access)

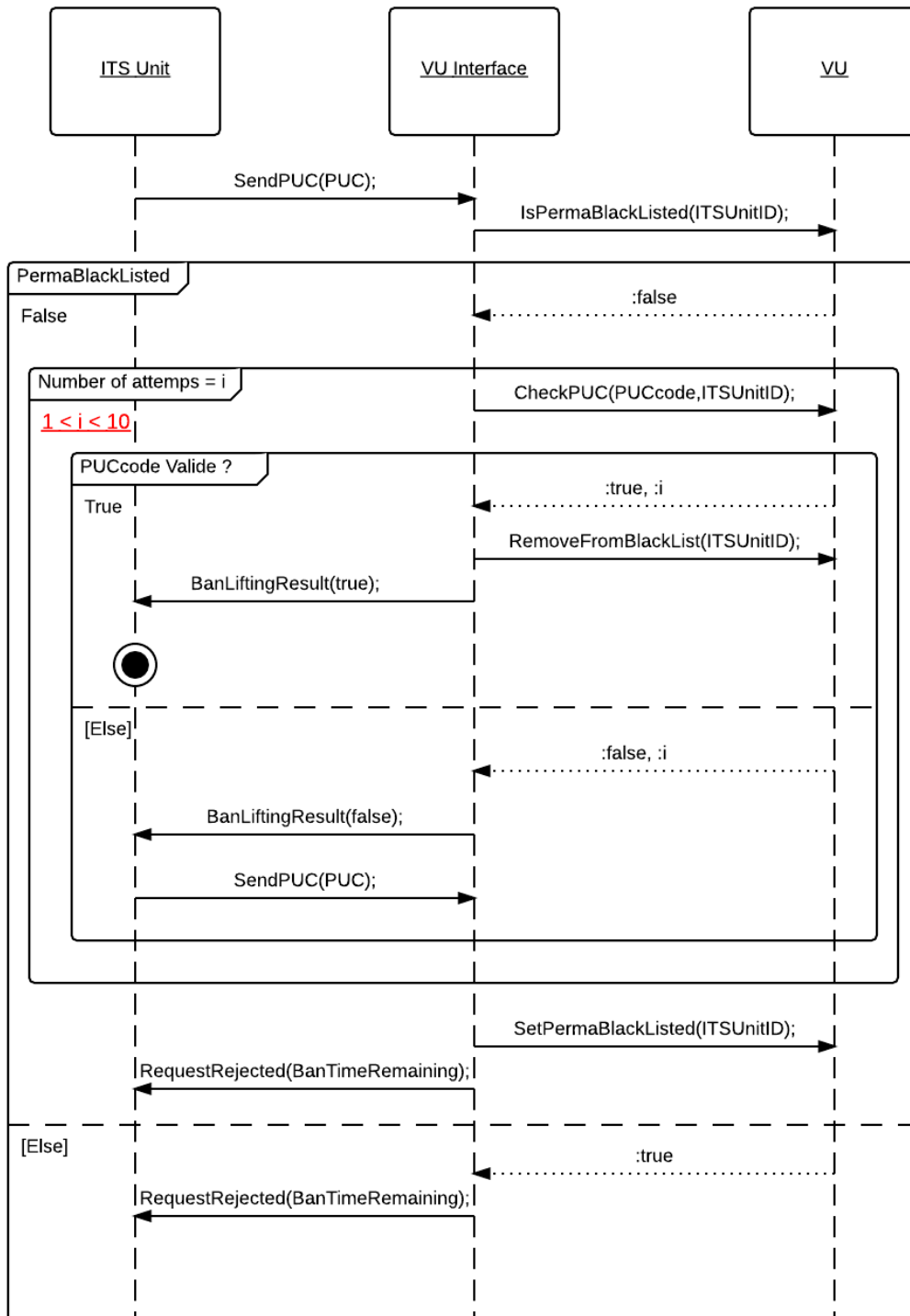


Figure 22. Sequence Diagram for PUC validation attempt

# ANNEX 3

## ASN.1 SPECIFICATIONS

FormatMessageModule DEFINITIONS AUTOMATIC TAGS ::= BEGIN

EXPORTS ;

IMPORTS SendPIN, SendPUC, PairingResult, RequestPIN, RequestRejected,

BanLiftingResult FROM PINPUCDataFieldsModule

RequestAccepted, RequestData, DataUnavailable FROM

RequestDataFieldsModule

SendITSID, NegativeAnswer FROM OtherDataFieldsModule;

CompleteMessage ::= SEQUENCE {

header Header,

data DataField,

checksum Checksum

}

-----  
--HEADER TYPES--  
-----

Header ::= SEQUENCE {

tgt IDList,

src IDList,

len BIT STRING (1..255)

}

vuID BIT STRING ::= 'EE'H

IDList ::= CHOICE {

vu BIT STRING (vuID),

```

itsUnits SEQUENCE OF BIT STRING,
    --Default hex Value:A0, redefined after first message exchange--
    --Each ID will be linked to the Bluetooth ID of the device--
    ...
}

-----
--DATAFIELDS TYPES--
-----

DataField ::=SEQUENCE{
    sid BIT STRING,
    trtp BIT STRING,
    subMBytes SubMessageBytes,
    dataField Content,
    ...
}

SubMessageBytes ::= SEQUENCE{
    currentSubM BIT STRING,
    totalSubM BIT STRING
}

Content ::= CHOICE{
    requestPIN RequestPIN,
    sendITSID SendITSID,
    sendPin SendPIN,
    pairRslt PairingResult,
    sendPUC SendPUC,
    banlift BanLiftingResult,
    requestRejected RequestRejected,
    requestData RequestData,
    requestOK RequestAccepted,
    dataUnavailable DataUnavailable,
    negAns NegativeAnswer
}

```

```
}
```

```
-----  
--CHECKSUM TYPES--  
-----
```

```
Checksum ::= SEQUENCE {  
    --SHA2 checksum  
}
```

```
END
```

```
PINPUCDataFieldsModule DEFINITIONS AUTOMATIC TAGS ::= BEGIN
```

```
EXPORTS SendPIN, SendPUC, PairingResult, RequestPIN, RequestRejected, BanLiftingResult;
```

```
IMPORTS ;
```

```
-----  
---Utils--  
-----
```

```
PUC ::= SEQUENCE (SIZE(8)) OF  
INTEGER (SIZE(0..9))
```

```
PIN ::= SEQUENCE (SIZE(4)) OF  
INTEGER (SIZE(0..9))
```

```
-----  
--Messages From ITS Unit--  
-----
```

```
SendPIN { PIN:pin } ::= SEQUENCE {  
    sid BIT STRING ('03'H),  
    pin PIN (pin)  
}
```

```
SendPUC { PUC:puc } ::= SEQUENCE {  
    sid BIT STRING ('05'H),
```

```

    puc PUC (puc)
}
-----
--Messages From VU--
-----

PairingResult ::= SEQUENCE{
    sid BIT STRING (04'H),
    result BOOLEAN
}

RequestPIN {MType:receivedRequest} ::= SEQUENCE{
    sid BIT STRING (01'H)
}

RequestRejected ::= SEQUENCE{
    sid BIT STRING (07'H),
    banTimeRemaining GeneralizedTime, --PermaBan == 1k years--    }

BanLiftingResult ::= SEQUENCE{
    sid BIT STRING (06'H),
    result BOOLEAN
}

END

RequestDataFields DEFINITIONS AUTOMATIC TAGS ::= BEGIN

EXPORTS RequestAccepted, RequestData, DataUnavailable ;
IMPORTS StandardEvent, PersonalEvent, StandardFault FROM EventsModule;

-----
---From ITS Unit--
-----

RequestData ::= SEQUENCE{
    sid BIT STRING (08'H),
    requestedData DataTypeCode,

```

```
    ...
}

-----
--From VU--
-----

RequestAccepted ::=SEQUENCE{
    sid BIT STRING ('09'H),
    trtp DataTypeCode,
    dataSheet CHOICE{
        standardData StandardTachDataContent,
        personalData PersonalTachDataContent,
        gnss GNSSDataContent,
        standardEvent StandardEventContent,
        personalEvent PersonalEventContent,
        standardFault StandardFaultContent,
        manufacturerdata ManufacturerDataContent,
        ...
    }
}
```

```
DataTypeCode ::=CHOICE{
    standardTachData BIT STRING ('01'H),
    personalTachData BIT STRING ('02'H),
    gnssData BIT STRING ('03'H),
    standardEventData BIT STRING ('04'H),
    personalEventData BIT STRING ('05'H),
    standardFaultData BIT STRING ('06'H),
    manufacturerData BIT STRING ('07'H),
    ...
}
```

```
DataUnavailable ::=SEQUENCE{
    sid BIT STRING ('0A'H),
```



```

    trtp DataTypeCode,
    reason UnavailableDataCodes
}

```

```

UnavailableDataCodes ::= CHOICE{
    noDataAvailable BIT STRING ('10'H),
    personalDataNotShared BIT STRING ('11'H),
    ...
}

```

-----  
 --Complete Tachograph Data--  
 -----

--The format of the data was taken from the ISO16844-7 norm, more information available in this ISO document--

```

Time ::= SEQUENCE{
    seconds INTEGER (0..59.75), --increment: 0.25s--
    minutes INTEGER (0..59), --increment: 1min--
    hours INTEGER (0..23), --increment: 1h--
    day INTEGER (0.25.. 31.75), --increment: 0.25d--
    month INTEGER (1..12), --increment: 1month--
    year INTEGER (1985..2235), --increment: 1year--
    locMinOffset INTEGER (-59..59), --increment: 1min--
    locHouroffset INTEGER (-23..23)--increment: 1h--
}

```

```

Date ::= SEQUENCE{
    month INTEGER (1..12), --increment: 1month--
    day INTEGER (0.25.. 31.75), --increment: 0.25d--
    year INTEGER (1985..2235) --increment: 1year--
}

```

```

DriverName ::=SEQUENCE{
    codePageSurname UTF8String, --See ISO/IEC 8859--

```

```
    surname UTF8String,  
    codePageFirstname UTF8String, --See ISO/IEC 8859--  
    firstname UTF8String,  
}
```

```
-----  
--Message Content--  
-----
```

```
StandardTachDataContent ::= SEQUENCE{  
    trtp DataTypeCode (DataTypeCode.&standardTachData),  
    personal BOOLEAN (FALSE),  
    data StandardTachyDataSheet,  
}
```

```
PersonalTachDataContent ::= SEQUENCE{  
    trtp DataTypeCode (DataTypeCode.&personalTachData),  
    personal BOOLEAN (TRUE),  
    data PersonalTachyDataSheet  
}
```

```
GNSSDataContent ::= SEQUENCE{  
    trtp DataTypeCode (DataTypeCode.&gnssData),  
    personal BOOLEAN (TRUE),  
    data GNSSDataSheet  
}
```

```
StandardEventContent ::= SEQUENCE{  
    trtp DataTypeCode (DataTypeCode.&standardEventData),  
    personal BOOLEAN (FALSE),  
    data StandardEventDataSheet  
}
```

```
PersonalEventContent ::= SEQUENCE{
```

```

    trtp DataTypeCode (DataTypeCode.&personalEventData),
    personal BOOLEAN (TRUE),
    data PersonalEventDataSheet
}

```

```

StandardFaultContent ::= SEQUENCE{
    trtp DataTypeCode (DataTypeCode.&standardFaultData),
    personal BOOLEAN (FALSE),
    data StandardFault
}

```

```

ManufacturerDataContent ::= SEQUENCE{
    trtp DataTypeCode (DataTypeCode.&manufacturerData),
    personal BOOLEAN (TRUE),
    ...
}

```

```

-----
--DATA SHEETS--
-----

```

--Data sheet format follows ISO 16844-7.--

```

StandardTachyDataSheet ::= SEQUENCE{
    vin UTF8String (SIZE(17)),
    calibrationDate Date,
    driveRecognize INTEGER (2 UNION 12),
    driverCardDriver1 INTEGER (2 UNION 12),
    driverCardDriver2 INTEGER (2 UNION 12),
    timeDate Time,
    highResolutionTotalVehicleDistance INTEGER (0..21055406), --increment: 5m--
    serviceComponentIdentification INTEGER (0..255),
    serviceDelayCalendarTimeBased INTEGER (-125..125), --increment: 1 week--
    nextCalibrationDate Date,
    speedAuthorised INTEGER (0..250.996), --increment 1/256km/h--
}

```

```

tachographCardSlot1 INTEGER (0..4...), --Maximum 250--
tachographCardSlot2 INTEGER (0..4...), --Maximum 250--
outOfScopeCondition INTEGER(2 UNION 12),
modeOfOperation INTEGER (0..4...), --Maximum 250--
registeringMemberState UTF8String,          vehicleRegistrationNumber SEQUENCE {
    codePageVRN INTEGER (0..255),
    vrn OCTET STRING (SIZE(13)),
},
tachographNextMandatoryDownloadDate Date,
...
}

```

PersonalTachyDataSheet ::= SEQUENCE{

```

    tachographVehicleSpeed INTEGER (0..250.996), --increment 1/256km/h--
    driver1WorkingState INTEGER (2 UNION 12 UNION 102 UNION 112 UNION 1002 UNION 1012...),
    driver2WorkingState INTEGER (2 UNION 12 UNION 102 UNION 112 UNION 1002 UNION 1012...),

    driver1TimeRelatedStates INTEGER(2 UNION 12 UNION 102 UNION 112 UNION 1002 UNION
        1012 UNION 1102 UNION 1112 UNION 10002 UNION 10012 UNION
        10102 UNION 10112 UNION 11002 UNION 11012...),
    driver2TimeRelatedStates INTEGER(2 UNION 12 UNION 102 UNION 112 UNION 1002 UNION
        1012 UNION 1102 UNION 1112 UNION 10002 UNION 10012 UNION
        10102 UNION 10112 UNION 11002 UNION 11012...),

    overSpeed INTEGER (2 UNION 12),
    driver1Identification INTEGER (SIZE(19)), --TODO NEED FURTHER SPECS FROM TACHO
REGULATION--
    driver2Identification INTEGER (SIZE(19)), --TODO NEED FURTHER SPECS FROM TACHO
REGULATION--
    driver1ContinuousDrivingTime INTEGER (0.. 64255), --increment: 1min--
    driver2ContinuousDrivingTime INTEGER (0.. 64255), --increment: 1min--
    driver1CurrentDurationOfSelectedActivity INTEGER (0.. 64255), --increment: 1min--
    driver2CurrentDurationOfSelectedActivity INTEGER (0.. 64255), --increment: 1min--
    driver1Name DriverName,
    driver2Name DriverName,

```

driver1CumulatedDrivingTimePreviousAndCurrentWeek INTEGER (0.. 64255), --increment: 1min--  
driver2CumulatedDrivingTimePreviousAndCurrentWeek INTEGER (0.. 64255), --increment: 1min--  
engineSpeed INTEGER(0..8031.875), --increment: 0,125r/min--  
driver1EndOfLastDailyRestPeriod Time,  
driver2EndOfLastDailyRestPeriod Time,  
driver1EndOfLastWeeklyRestPeriod Time,  
driver2EndOfLastWeeklyRestPeriod Time,  
driver1EndOfSecondLastWeeklyRestPeriod Time,  
driver2EndOfSecondLastWeeklyRestPeriod Time,  
driver1CurrentDailyDrivingTime INTEGER (0.. 64255), --increment: 1min--  
driver2CurrentDailyDrivingTime INTEGER (0.. 64255), --increment: 1min--  
driver1CurrentWeeklyDrivingTime INTEGER (0.. 64255), --increment: 1min--  
driver2CurrentWeeklyDrivingTime INTEGER (0.. 64255), --increment: 1min--  
driver1TimeLeftUntilNewDailyRestPeriod INTEGER (0.. 64255), --increment: 1min--  
driver2TimeLeftUntilNewDailyRestPeriod INTEGER (0.. 64255), --increment: 1min--  
driver1CardExpiryDate Date,  
driver2CardExpiryDate Date,  
driver1CardNextMandatoryDownloadDate Date,  
driver2CardNextMandatoryDownloadDate Date,  
driver1TimeLeftUntilNewWeeklyRestPeriod INTEGER (0.. 64255), --increment: 1min--  
driver2TimeLeftUntilNewWeeklyRestPeriod INTEGER (0.. 64255), --increment: 1min--  
driver1NumberOfTimes9hDailyDrivingTimesExceeded INTEGER (0..13),  
driver2NumberOfTimes9hDailyDrivingTimesExceeded INTEGER (0..13),  
driver1CumulativeUninterruptedRestTime INTEGER (0.. 64255), --increment: 1min--  
driver2CumulativeUninterruptedRestTime INTEGER (0.. 64255), --increment: 1min--  
driver1MinimumDailyRest INTEGER (0.. 64255), --increment: 1min--  
driver2MinimumDailyRest INTEGER (0.. 64255), --increment: 1min--  
driver1MinimumWeeklyRest INTEGER (0.. 64255), --increment: 1min--  
driver2MinimumWeeklyRest INTEGER (0.. 64255), --increment: 1min--  
driver1MaximumDailyPeriod INTEGER (0..250), --increment: 1h--  
driver2MaximumDailyPeriod INTEGER (0..250), --increment: 1h--  
driver1MaximumDailyDrivingTime INTEGER (910 UNION 1010),  
driver2MaximumDailyDrivingTime INTEGER (910 UNION 1010),  
driver1NumberOfUsedReducedDailyRestPeriods INTEGER (0..13),

```
driver2NumberOfUsedReducedDailyRestPeriods INTEGER (0..13),
driver1RemainingCurrentDrivingTime INTEGER (0.. 64255), --increment: 1min--
driver2RemainingCurrentDrivingTime INTEGER (0.. 64255), --increment: 1min--
...
}
```

```
GNSSDataSheet ::= SEQUENCE {
  gnssPosition GeoCoordinates
  --See Appendix 1 for definition of GeoCoordinates--
}
```

```
StandardEventDataSheet ::= SEQUENCE{
  events SEQUENCE OF StandardEvent
}
```

```
PersonalEventDataSheet ::= SEQUENCE{
  events SEQUENCE OF PersonalEvent
}
```

**END**

**EventsModule DEFINITIONS** AUTOMATIC TAGS ::= **BEGIN**

**EXPORTS** ALL;

**IMPORTS** NationAlpha **FROM** Appendix1; --See Appendix 1 for more information about NationAlpha--

```
SecurityBreachEvent ::=SEQUENCE{
  --See Annex 1B for more information--
}
```

```
RecordingEquipmentFaultType ::= SEQUENCE{
  --See Annex 1B for more information--
}
```

```
StandardEvent ::= CHOICE{
```

```

insertionInvalidCard InsertionOfANonValidCard,
cardConflict CardConflict,
timeOverlap TimeOverlap,
previousSessionNotClosed LastCardSessionNotCorrectlyClosed,
overSpeeding OverSpeeding,
powerSupplyInterruption PowerSupplyInterruption,
comErrorWithRemoteFacility CommunicationErrorWithTheRemoteCommunicationFacility,
absenceGNSSPosition AbsenceOfPositionInformationFromGNSSReceiver,
positionDataError PositionDataError,
motionDataError MotionDataError,
vehicleMotionConflict VehicleMotionConflict,
securityBreachAttempt SecurityBreachAttempt,
timeConflict TimeConflict,
...
}

```

```

PersonalEvent ::= CHOICE{
    lackOfAppropriateCard DrivingWithoutAnAppropriateCard,
    cardInsertionWhileDriving CardInsertionWhileDriving,
    overSpeeding OverSpeeding,
    ...
}

```

```

StandardFault ::= CHOICE{
    cardFault CardFault,
    recordingEquipmentFault RecordingEquipmentFault,
    ...
}

```

```

-----
--EVENTS LIST--
-----

```

```

InsertionOfANonValidCard ::= SEQUENCE{

```

```
beginDate GeneralizedTime,  
endDate GeneralizedTime,  
carsdType SEQUENCE OF UTF8String,  
cardsNumber SEQUENCE OF INTEGER,  
issuingMemberState SEQUENCE OF NationAlpha,  
cardsGeneration SEQUENCE OF INTEGER  
}
```

```
CardConflict ::= SEQUENCE{  
beginDate GeneralizedTime,  
endDate GeneralizedTime,  
carsdType SEQUENCE OF UTF8String,  
cardsNumber SEQUENCE OF INTEGER,  
issuingMemberState SEQUENCE OF NationAlpha,  
cardsGeneration SEQUENCE OF INTEGER  
}
```

```
TimeOverlap ::=SEQUENCE{  
beginDate GeneralizedTime,  
endDate GeneralizedTime,  
carsdType SEQUENCE OF UTF8String,  
cardsNumber SEQUENCE OF INTEGER,  
issuingMemberState SEQUENCE OF NationAlpha,  
cardsGeneration SEQUENCE OF INTEGER,  
numberSimilarEvent INTEGER  
}
```

```
DrivingWithoutAnAppropriateCard ::= SEQUENCE{  
beginDate GeneralizedTime,  
endDate GeneralizedTime,  
carsdType SEQUENCE OF UTF8String,  
cardsNumber SEQUENCE OF INTEGER,  
issuingMemberState SEQUENCE OF NationAlpha,  
cardsGeneration SEQUENCE OF INTEGER,
```



```

        numberOfSimilarEvent INTEGER
    }

CardInsertionWhileDriving ::= SEQUENCE {
    date GeneralizedTime,
    carsdType SEQUENCE OF UTF8String,
    cardsNumber SEQUENCE OF INTEGER,
    issuingMemberState SEQUENCE OF NationAlpha,
    numberOfSimilarEvents INTEGER
}

LastCardSessionNotCorrectlyClosed ::= SEQUENCE {
    beginDate GeneralizedTime,
    endDate GeneralizedTime,
    carsdType SEQUENCE OF UTF8String,
    cardsNumber SEQUENCE OF INTEGER,
    issuingMemberState SEQUENCE OF NationAlpha,
    cardsGeneration SEQUENCE OF INTEGER,
    oldSession SEQUENCE {
        beginDate GeneralizedTime,
        endDate GeneralizedTime,
        vrn UTF8String,
        issuingMemberState NationAlpha,
        cardsGeneration INTEGER,
    }
}

OverSpeeding ::= SEQUENCE {
    beginDate GeneralizedTime,
    endDate GeneralizedTime,
    maximumSpeed INTEGER,
    averageSpeed INTEGER,
    cardType UTF8String,
    cardNumber INTEGER,
}

```

```
    issuingMemberState NationAlpha,  
    cardGeneration INTEGER,  
    numberOfSimilarEvents INTEGER  
}
```

```
PowerSupplyInterruption ::=SEQUENCE{  
    beginDate GeneralizedTime,  
    endDate GeneralizedTime,  
    carsdType SEQUENCE OF UTF8String,  
    cardsNumber SEQUENCE OF INTEGER,  
    issuingMemberState SEQUENCE OF NationAlpha,  
    cardsGeneration SEQUENCE OF INTEGER,  
    numberOfSimilarEvent INTEGER  
}
```

```
CommunicationErrorWithTheRemoteCommunicationFacility ::=SEQUENCE{  
    beginDate GeneralizedTime,  
    endDate GeneralizedTime,  
    carsdType SEQUENCE OF UTF8String,  
    cardsNumber SEQUENCE OF INTEGER,  
    issuingMemberState SEQUENCE OF NationAlpha,  
    cardsGeneration SEQUENCE OF INTEGER,  
    numberOfSimilarEvent INTEGER  
}
```

```
AbsenceOfPositionInformationFromGNSSReceiver ::= SEQUENCE{  
    beginDate GeneralizedTime,  
    endDate GeneralizedTime,  
    carsdType SEQUENCE OF UTF8String,  
    cardsNumber SEQUENCE OF INTEGER,  
    issuingMemberState SEQUENCE OF NationAlpha,  
    cardsGeneration SEQUENCE OF INTEGER,  
    numberOfSimilarEvent INTEGER  
}
```

```
PositionDataError ::= SEQUENCE{
    beginDate GeneralizedTime,
    endDate GeneralizedTime,
    cardsType SEQUENCE OF UTF8String,
    cardsNumber SEQUENCE OF INTEGER,
    issuingMemberState SEQUENCE OF NationAlpha,
    cardsGeneration SEQUENCE OF INTEGER,
    numberOfSimilarEvent INTEGER
}
```

```
MotionDataError ::= SEQUENCE{
    beginDate GeneralizedTime,
    endDate GeneralizedTime,
    cardsType SEQUENCE OF UTF8String,
    cardsNumber SEQUENCE OF INTEGER,
    issuingMemberState SEQUENCE OF NationAlpha,
    cardsGeneration SEQUENCE OF INTEGER,
    numberOfSimilarEvent INTEGER
}
```

```
VehicleMotionConflict ::= SEQUENCE{
    beginDate GeneralizedTime,
    endDate GeneralizedTime,
    cardsType SEQUENCE OF UTF8String,
    cardsNumber SEQUENCE OF INTEGER,
    issuingMemberState SEQUENCE OF NationAlpha,
    cardsGeneration SEQUENCE OF INTEGER,
    numberOfSimilarEvent INTEGER
}
```

```
SecurityBreachAttempt ::= SEQUENCE{
    beginDate GeneralizedTime,
    endDate GeneralizedTime OPTIONAL,
}
```

```
cardsType SEQUENCE OF UTF8String,  
cardsNumber SEQUENCE OF INTEGER,  
issuingMemberState SEQUENCE OF NationAlpha,  
numberOfSimilarEvent INTEGER,  
typeOfEvent SecurityBreachEvent  
}
```

```
TimeConflict ::= SEQUENCE{  
    beginDate GeneralizedTime,  
    endDate GeneralizedTime,  
    cardsType SEQUENCE OF UTF8String,  
    cardsNumber SEQUENCE OF INTEGER,  
    issuingMemberState SEQUENCE OF NationAlpha,  
    cardsGeneration SEQUENCE OF INTEGER,  
    numberOfSimilarEvent INTEGER  
}
```

```
-----  
--FAULTS LIST--  
-----
```

```
CardFault ::= SEQUENCE{  
    beginDate GeneralizedTime,  
    endDate GeneralizedTime,  
    cardsType SEQUENCE OF UTF8String,  
    cardsNumber SEQUENCE OF INTEGER,  
    issuingMemberState SEQUENCE OF NationAlpha,  
    cardsGeneration SEQUENCE OF INTEGER,  
}
```

```
RecordingEquipmentFault ::= SEQUENCE{  
    beginDate GeneralizedTime,  
    endDate GeneralizedTime,  
    faultType RecordingEquipmentFaultType,
```

cardsType **SEQUENCE OF** UTF8String,  
cardsNumber **SEQUENCE OF** INTEGER,  
issuingMemberState **SEQUENCE OF** NationAlpha,  
cardsGeneration **SEQUENCE OF** INTEGER,

}

**END**

## APPENDIX 14. REMOTE COMMUNICATION FUNCTION

### TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION</b>	<b>560</b>
<b>2</b>	<b>SCOPE</b>	<b>561</b>
<b>3</b>	<b>ACRONYMS, DEFINITIONS AND NOTATIONS</b>	<b>562</b>
<b>4</b>	<b>OPERATIONAL SCENARIOS</b>	<b>564</b>
<b>4.1</b>	<b>Overview</b>	<b>564</b>
4.1.1	.. Preconditions to data transfer via 5.8 GHz DSRC interface	564
4.1.2	.. Profile 1a: via a hand aimed or temporary roadside mounted and aimed Remote Early Detection Communication Reader	565
4.1.3	.. Profile 1b: via a vehicle mounted and directed Remote Early Detection Communication Reader (REDCR) .....	566
<b>4.2</b>	<b>Security/Integrity</b>	<b>566</b>
<b>5</b>	<b>REMOTE COMMUNICATION DESIGN AND PROTOCOLS</b>	<b>567</b>
<b>5.1</b>	<b>Design</b>	<b>567</b>
<b>5.2</b>	<b>Workflow</b>	<b>571</b>
5.2.1	.. Operations	571
5.2.2	.. Interpretation of the Data received via the DSRC communication	572
<b>5.3</b>	<b>DSRC Physical interface parameters for remote communication</b>	<b>572</b>
5.3.1	.. Location constraints	572
5.3.2	.. Downlink and uplink parameters	572
5.3.3	.. Antenna design	577
<b>5.4</b>	<b>DSRC Protocol requirements for RTM</b>	<b>577</b>
5.4.1	.. Overview	577
5.4.2	.. Commands	581
5.4.3	.. Interrogation command sequence	581
5.4.4	.. Data structures	582
5.4.5	.. Elements of RtmData, actions performed and definitions	587
5.4.6	.. Data transfer mechanism	594
5.4.7	.. Detailed DSRC transaction description	594

---

5.4.8 .. DSRC Test transaction description	601
<b>5.5 Support for Directive 2015/71/EC</b>	<b>604</b>
5.5.1 .. Overview	604
5.5.2 .. Commands	604
5.5.3 .. Interrogation command sequence	604
5.5.4 .. Data structures	604
5.5.5 .. ASN.1 module for the OWS DSRC transaction	604
5.5.6 .. Elements of OwsData, actions performed and definitions	606
5.5.7 .. Data transfer mechanisms	607
<b>5.6 Data transfer between the DSRC-VU and VU</b>	<b>607</b>
5.6.1 .. Physical Connection and interfaces	607
5.6.2 .. Application Protocol	608
<b>5.7 Error handling</b>	<b>609</b>
5.7.1 .. Recording and communication of the Data in the DSRC-VU	609
5.7.2 .. Wireless Communication errors	609
<b>6 COMMISSIONING AND PERIODIC INSPECTION TESTS FOR THE REMOTE COMMUNICATION FUNCTION</b>	<b>611</b>
<b>6.1 General</b>	<b>611</b>
<b>6.2 ECHO</b>	<b>611</b>
<b>6.3 Tests to validate the secure data content</b>	<b>611</b>

## 1 Introduction

This Appendix specifies the design and the procedures to follow in order to perform the remote communication function (the Communication) as required in Article 9 of Regulation (EU) N°. 165/2014 (the Regulation).

DSC\_1 Regulation (EU) N°. 165/2014 determines that the tachograph shall be equipped with a remote communication functionality that shall enable agents of the competent control authorities to read tachograph information from passing vehicles by using remote interrogation equipment (the Remote early detection communication reader [REDCR]), specifically, interrogation equipment connecting wirelessly using CEN 5.8 GHz Dedicated Short Range Communication (DSRC) interfaces.

It is important to comprehend that this functionality is intended to serve only as a pre-filter in order to select vehicles for closer inspection, and it does not replace the formal inspection process as determined in the provisions of Regulation (EU) N°. 165/2014. See recital 9 in the preamble of this regulation, stating that remote communication between the tachograph and control authorities for roadside control purposes facilitates targeted roadside checks.

DSC\_2 *The Data* shall be exchanged using *the Communication* which shall be a wireless intercourse using 5.8 GHz DSRC wireless communications consistent with this Appendix and tested against the appropriate parameters of EN 300 674-1, {Electromagnetic compatibility and Radio spectrum Matters (ERM); Road Transport and Traffic Telematics (RTTT); Dedicated Short Range Communication (DSRC) transmission equipment (500 kbit/s / 250 kbit/s) operating in the 5,8 GHz Industrial, Scientific and Medical (ISM) band; Part 1: General characteristics and test methods for Road Side Units (RSU) and On-Board Units (OBU)}.

DSC\_3 *The Communication* shall be established with the communications equipment only when so requested by the equipment of the competent control authority using compliant radio-communication means (*the Remote early detection communication reader (REDCR)*).

DSC\_4 *The Data* shall be secured to ensure integrity.

DSC\_5 Access to *the Data* communicated shall be restricted to competent control authorities authorised to check infringements of Regulation (EC) N°. 561/2006 and of Regulation (EU) N°. 165/2014 and to workshops in so far as it is necessary to verify the correct functioning of the tachograph.

DSC\_6 *The Data* exchanged during *the Communication* shall be limited to the data necessary for the purpose of targeting roadside checks of vehicles with a potentially manipulated or misused tachograph.

DSC\_7 Data integrity and security shall be obtained by securing *the Data* within the Vehicle Unit (VU) and by passing only the secured payload data and security related data (see 4.4.4) across the wireless 5.8 GHz DSRC remote communication medium, meaning that only authorised persons of competent control authorities have the means to understand the data passed across *the Communication* and to verify its authenticity. See Appendix 11 Common Security Mechanisms.

DSC\_8 *The Data* shall contain a timestamp for the time of its last update.



- DSC\_9 The content of the security data shall be known only to and within the control of the competent control authorities, and those parties with whom they share this information and is outwith the provisions of *the Communication* that is the subject of this Appendix, save that *the Communication* makes provision to transfer a packet of security data with every packet of payload data.
- DSC\_10 The same architecture and equipment shall be capable be used to acquire other data concepts (such as weigh-on-board) using the architecture specified herein.
- DSC\_11 For clarification, in accordance with the provisions of Regulation (EU) N°. 165/2014 (Article 7), data concerning the identity of the driver shall not be communicated across *the Communication*.

## 2 Scope

The scope of this Appendix is to specify how agents of the competent control authorities use a specified 5.8 GHz DSRC wireless communication to remotely obtain data (*the Data*) from a targeted vehicle that identifies that the targeted vehicle is in potential violation of Regulation (EU) N°. 165/2014 and should be targeted for consideration to be stopped for further investigation.

Regulation (EU) N°. 165/2014 requires that the Data collected shall be limited to data or pertaining to data that identifies a potential infringement, as defined in Article 9 of Regulation (EU) N°. 165/2014.

In this scenario, the time available for communication is limited, because *the Communication* is targeted and of a short- range design. Further, the same communication means for remote tachograph monitoring (RTM) may also be used by the competent control authorities for other applications (such as the maximal weights and dimensions for heavy goods vehicles defined in Directive 2015/719/EC) and such operations may be separate or sequential at the discretion of the competent control authorities.

This Appendix specifies:

- The communications equipment, procedures and protocols to be used for *the Communication*
- The Standards and Regulations to which the radio equipment shall comply
- The presentation of *the Data* to *the Communication* equipment
- The enquiry and download procedures and sequence of operations
- *The Data* to be transferred
- Potential interpretation of *the Data* transferred across *the Communication*
- The provisions for security data relating to *the Communication*
- The availability of *the Data* to the competent control authorities
- How the *Remote early detection communication reader* can request different freight and fleet data concepts

For clarification, this Appendix does not specify:

- the collection of *the Data* operation and management within the VU (which shall be a function of product design unless specified elsewhere within Regulation (EU) N°. 165/2014)
- the form of presentation of collected data to the agent of the competent control authorities, nor the criteria which shall be used by the competent control authorities to decide which vehicles to stop (which shall be a function of product design unless specified elsewhere within Regulation (EU) N°. 165/2014 or a policy decision of the competent control authorities). For clarification: *the Communication* only makes *the Data* available to the competent control authorities in order that they may make informed decisions
- Data security provisions (such as encryption) concerning the content of *the Data* (which shall be specified within Appendix 11 Common Security Mechanisms).
- detail of any data concepts other than RTM which may be obtained using the same architecture and equipment
- detail of the behaviour and management between VU's and the DSRC-VU, nor the behaviour within the DSRC-VU (other than to provide *the Data* when so requested by an REDCR).

### 3 Acronyms, definitions and notations

The following acronyms and definitions specific to this Appendix are used in this appendix:

<b><i>the Antenna</i></b>	electrical device which converts electric power into radio waves, and vice versa used in combination with a radio transmitter or radio receiver. In operation, a radio transmitter supplies an electric current oscillating at radio frequency to the antenna's terminals, and the antenna radiates the energy from the current as electromagnetic waves (radio waves). In reception, an antenna intercepts some of the power of an electromagnetic wave in order to produce a tiny voltage at its terminals, that is applied to a receiver to be amplified
<b><i>the Communication</i></b>	exchange of information/data between a DSRC-REDCR and a DSRC-VU according to section 4 in a master-slave relationship to obtain the Data.
<b><i>the Data</i></b>	secured data of defined format (see 4.4.4) requested by the <i>DSRC-REDCR</i> and provided to the <i>DSRC-REDCR</i> by the <i>DSRC-VU</i> across a 5.8 GHz DSRC link as defined in 5 below
<b><i>Regulation (EC) No. 165/2014</i></b>	Regulation (EU) N°. 165/2014 of the European Parliament and of the Council of 4 February 2014 on tachographs in road transport, repealing Council Regulation (EEC) N°. 3821/85 on recording equipment in road transport and amending Regulation (EC) N°. 561/2006 of the European Parliament and of the Council on the harmonisation of certain social legislation relating to road transport

<b>AID</b>	Application Identifier
<b>BLE</b>	Bluetooth Low Energy

<b>BST</b>	Beacon Service Table
<b>CIWD</b>	Card insertion while driving
<b>CRC</b>	cyclic redundancy check
<b>DSC (n)</b>	identifier of a requirement for a specific DSRC appendix
<b>DSRC</b>	Dedicated Short Range Communication
<b>DSRC-REDCR</b>	DSRC – Remote Early Detection Communication Reader.
<b>DSRC-VU</b>	DSRC – Vehicle Unit. This is the “remote early detection facility” defined in Annex 1C.
<b>DWVC</b>	Driving without valid card
<b>EID</b>	Element Identifier
<b>LLC</b>	Logical Link Control
<b>LPDU</b>	LLC Protocol Data Unit
<b>OWS</b>	Onboard Weighing System
<b>PDU</b>	Protocol Data Unit
<b>REDCR</b>	Remote early detection communication reader. This is the “remote early detection communication reader equipment” defined in Annex 1C.
<b>RTM</b>	Remote Tachograph Monitoring
<b>SM-REDCR</b>	Security Module-Remote early detection communication reader
<b>TARV</b>	Telematics Applications for Regulated Vehicles (ISO 15638 series of Standards)
<b>VU</b>	Vehicle Unit
<b>VUPM</b>	Vehicle Unit Payload Memory
<b>VUSM</b>	Vehicle Unit Security Module
<b>VST</b>	Vehicle Service Table
<b>WIM</b>	Weigh in motion
<b>WOB</b>	Weigh on board

The specification defined in this Appendix refers to and depends upon all or parts of the following regulations and standards. Within the clauses of this Appendix the relevant standards, or relevant clauses of standards, are specified. In the event of any contradiction the clauses of this Appendix shall take precedence. In the event of any contradiction where no specification is clearly determined in this Appendix, operating within ERC 70-03 (and tested against the appropriate parameters of EN 300 674-1) shall take precedence, followed in descending order of preference by EN 12795, EN 12253 EN 12834 and EN 13372, 6.2, 6.3, 6.4 and 7.1.

Regulations and standards referenced in this Appendix are:

- [1] Regulation (EU) N°. 165/2014 of the European Parliament and of the Council of 4 February 2014 on tachographs in road transport, repealing Council Regulation (EEC) N°. 3821/85 on recording equipment in road transport and amending Regulation (EC) N°. 561/2006 of the European Parliament and of the Council on the harmonisation of certain social legislation relating to road transport.
- [2] Regulation (EC) N°. 561/2006 of the European Parliament and of the Council of 15 March 2006 on the harmonisation of certain social legislation relating to road transport and amending Council Regulations

- (EEC) N°. 3821/85 and (EC) N°. 2135/98 and repealing Council Regulation (EEC) N°. 3820/85 (Text with EEA relevance).
- [3] ERC 70-03 CEPT: ECC Recommendation 70-03 : Relating to the Use of Short Range Devices (SRD)
  - [4] ISO 15638 Intelligent transport systems — Framework for cooperative telematics applications for regulated commercial freight vehicles (TARV).
  - [5] EN 300 674-1 Electromagnetic compatibility and Radio spectrum Matters (ERM); Road Transport and Traffic Telematics (RTTT); Dedicated Short Range Communication (DSRC) transmission equipment (500 kbit/s / 250 kbit/s) operating in the 5,8 GHz Industrial, Scientific and Medical (ISM) band; Part 1: General characteristics and test methods for Road Side Units (RSU) and On-Board Units (OBU).
  - [6] EN 12253 Road transport and traffic telematics - Dedicated short-range communication - Physical layer using microwave at 5.8 GHz.
  - [7] EN 12795 Road transport and traffic telematics - Dedicated short-range communication - Data link layer: medium access and logical link control.
  - [8] EN 12834 Road transport and traffic telematics - Dedicated short-range communication - Application layer.
  - [9] EN 13372 Road transport and traffic telematics - Dedicated short-range communication - Profiles for RTTT applications
  - [10] ISO 14906 Electronic fee collection — Application interface definition for dedicated short- range communication

## Operational Scenarios

### 3.1 Overview

Regulation (EU) N°. 165/2014 provides specific and controlled scenarios within which *the Communication* is to be used.

The scenarios supported are:

*“Communication Profile 1: Roadside inspection using a short range wireless communication Remote Early Detection Communication Reader instigating a physical roadside inspection (master-:-slave)*

*Reader Profile 1a: via a hand aimed or temporary roadside mounted and aimed Remote Early Detection Communication*

*Reader Profile 1b: via a vehicle mounted and directed Remote Early Detection Communication Reader”.*

#### 3.1.1 Preconditions to data transfer via 5.8 GHz DSRC interface

NOTE: In order to understand the context of the preconditions the reader is referred to Figure 14.3 below.

##### 3.1.1.1 Data held in VU

DSC\_12 The VU shall be responsible to keep updated every 60 seconds and maintain the data to be stored in the VU, without any involvement of the DSRC communication function. The means by which this is achieved is internal to the VU, specified in Regulation (EU) N°. 165/2014, Annex 1 C, section 3.19 “*Remote communication for targeted roadside checks*” and is not specified in this Appendix.

### 3.1.1.2 Data provided to DSRC-VU Facility

- DSC\_13 The VU shall be responsible to update the DSRC tachograph data (*the Data*) whenever the data stored in the VU is updated at the interval determined in 4.1.1.1 (DSC\_12), without any involvement of the DSRC communication function.
- DSC\_14 The VU data shall be used as a basis to populate and update *the Data*, the means by which this is achieved, is specified in Annex 1.C, section 3.19 “*Remote communication for targeted roadside checks*” or if there is no such specification it is a function of product design and is not specified in this Appendix. For the design of the connection between DSRC-VU facility and the VU, please refer to section 5.6.

### 3.1.1.3 Content of the Data

- DSC\_15 The content and format of *the Data* shall be such that, once decrypted, it shall be structured and made available in the form and format specified in 5.4.4 of this Appendix (Data structures).

### 3.1.1.4 Data presentation

- DSC\_16 *The Data*, having been kept frequently updated in accordance with the procedures determined in 4.1.1.1, shall be secured prior to presentation to the *DSRC-VU*, and presented as a secured data concept value, for temporary storage in the *DSRC-VU* as the current version of *the Data*. This data is transferred from the *VUSM* to the DSRC function *VUPM*. The *VUSM* and *VUPM* are functions and not necessarily physical entities. The form of physical instantiation to perform these functions shall be a matter of product design unless specified elsewhere in Regulation (EU) N°. 165/2014.

### 3.1.1.5 Security data

- DSC\_17 Security data (*securityData*), comprising the data required by the *REDCR* to complete its ability to decrypt *the Data* shall be supplied as defined in Appendix 11 Common Security Mechanisms and presented as a data concept value, for temporary storage in the *DSRC-VU* as the current version of *securityData*, in the form defined in this Appendix section 5.4.4.

### 3.1.1.6 VUPM data available for transfer across the DSRC interface

- DSC\_18 The data concept which shall always be available in the DSRC function *VUPM* for immediate transfer upon request by the *REDCR* is defined in section 5.4.4 for full ASN.1 Module specifications.

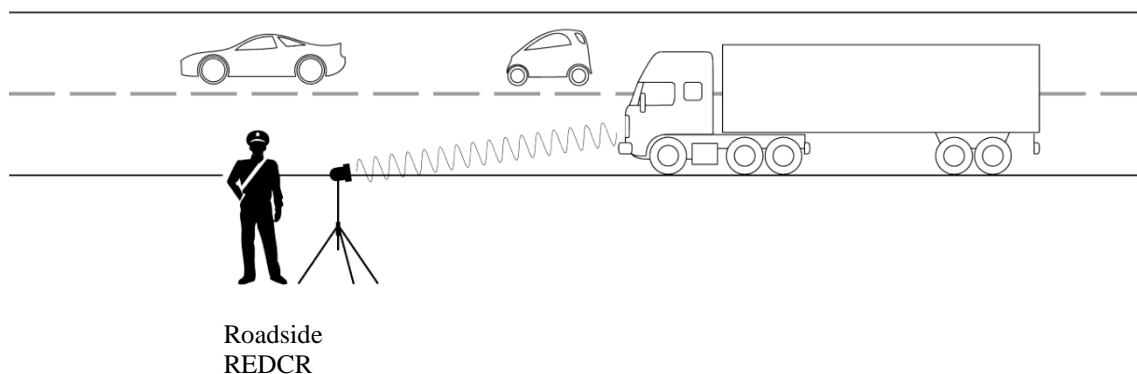
## General overview of communication Profile 1

This profile covers the use case where an agent of the competent control authorities, uses a short range remote communication Remote Early Detection Communication Reader (5.8 GHz DSRC interfaces operating within ERC 70-03, and tested against the appropriate parameters of EN 300 674-1 as described in section 4) (*the REDCR*) to remotely identify a vehicle which is potentially in violation of Regulation (EU) N°. 165/2014. Once identified, the agent of the competent control authorities who is controlling the interrogation decides whether the vehicle should be stopped.

### 3.1.2 Profile 1a: via a hand aimed or temporary roadside mounted and aimed Remote Early Detection Communication Reader

In this use case the agent of the competent control authorities is situated at the roadside, and aims a hand held, tripod mounted, or similar portable, *REDCR* from the roadside towards the centre of the windshield of the targeted vehicle. The interrogation is made using 5.8 GHz DSRC interfaces operating within ERC 70-03, and tested against the appropriate parameters of EN 300 674-1 as described in section 4. See Figure 14.1 (Use Case 1).

## Use case 1

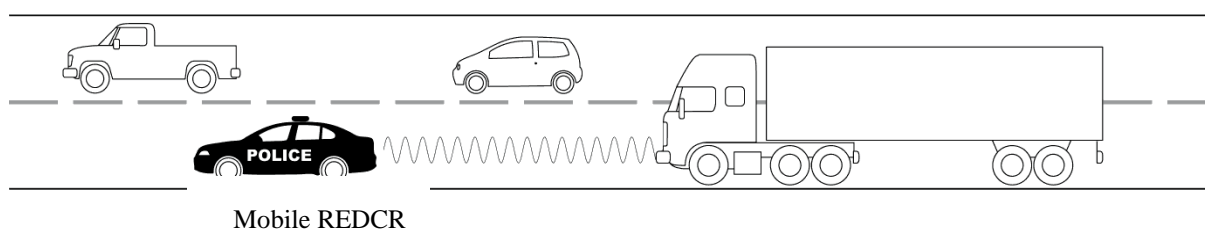


**Figure 14.1 — Roadside interrogation using 5.8 GHz DSRC**

### 3.1.3 Profile 1b: via a vehicle mounted and directed Remote Early Detection Communication Reader (REDCR)

In this use case the agent of the competent control authorities is situated within a moving vehicle, and either aims a hand held, portable *REDCR* from the vehicle towards the centre of the windshield of the targeted vehicle, or *the REDCR* is mounted within or on the vehicle so as to point towards the centre of the windshield of the targeted vehicle when the Remote Early Detection Communication Reader's vehicle is in a particular position relevant to the targeted vehicle (for example directly ahead in a stream of traffic). The interrogation is made using 5.8 GHz DSRC interfaces operating within ERC 70-03, and tested against the appropriate parameters of EN 300 674-1 as described in section 4. See Figure 14.2. (Use Case 2).

## Use case 2



**Figure 14.2 — Vehicle based interrogation using 5.8 GHz DSRC**

## 3.2 Security/Integrity

To give the possibility to verify the authenticity and integrity of downloaded data through the remote communication, the secured *Data* is verified and decrypted in accordance with Appendix 11 Common Security Mechanisms.

4 Remote Communication design and protocols

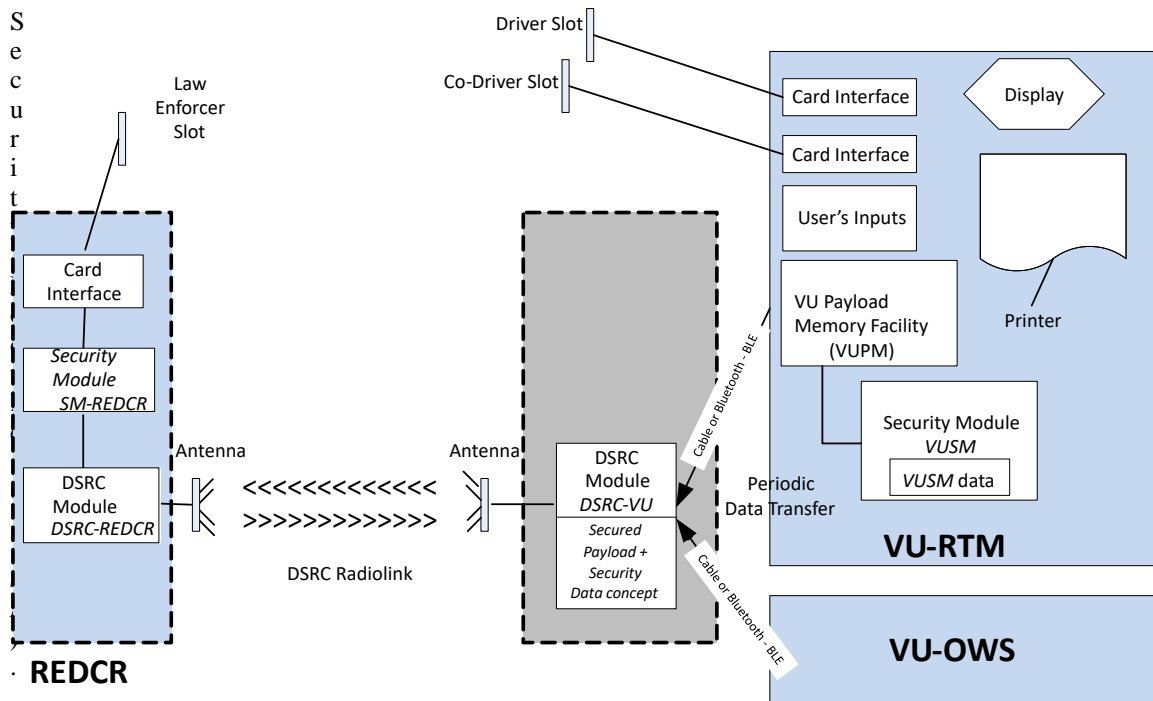
4.1 Design

The design of the remote communication function in the Smart Tachograph is shown as described in Figure 14.3.

**Figure 14.3 — Design of the remote communication function**

DSC\_19 The following functions are located in the VU:

- S  
e  
c  
u  
r  
i  
t  
y



T

his function present in the VU is responsible for securing the Data which is to be transmitted from the DSRC-VU to the agent of the competent control authorities via remote communication.

- The secured data is stored in the VUSM memory. At intervals determined in 4.1.1.1 (DSC\_12), the VU encrypts and replenishes the RTMdata concept (which comprises payload data and security data concept values determined below in this Appendix) held in the memory of the DSRC-VU. The operation of the security module is defined in Appendix 11 Common Security Mechanisms and outwith the scope of this Appendix, save that it shall be required to provide updates to the VU Communication facility each time the VUSM data changes.
- The communication between the VU and the DSRC-VU may be a wired communication or a Bluetooth Low Energy (BLE) communication, and the physical location of the DSRC-VU may be integral with the antenna on the windshield of the vehicle, may be internal to the VU, or located somewhere between.
- The DSRC-VU shall have a reliable source of power available at all times. The means by which it is provided with its power is a design decision.

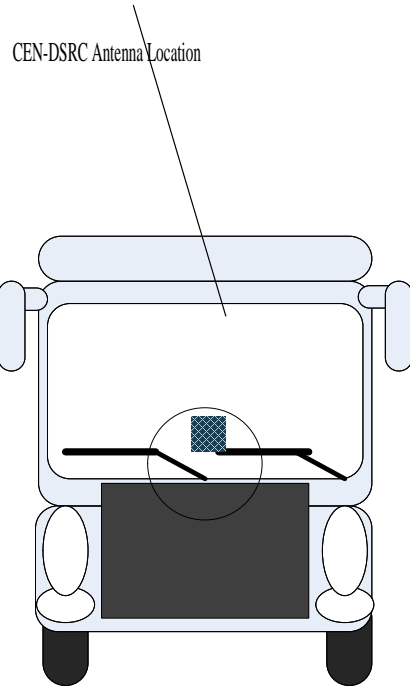
- The memory of the DSRC-VU shall be non-volatile in order to maintain the Data in the DSRC-VU even when the vehicle ignition is switched off.
- If the communication between the VU and the DSRC-VU is made via BLE and the power source is a non-recharging battery, the power source of the DSRC-VU shall be replaced at every Periodic Inspection, and the manufacturer of the DSRC-VU equipment shall be responsible to ensure that the power supply is adequate to last from one Periodic Inspection to the next Periodic Inspection, maintaining normal access to the data by an REDCR throughout the period without failure or interruption.
- VU RTM ‘payload memory’ facility (*VUPM*). This function present in the VU is responsible for providing and updating *the Data*. The content of *The Data*. (“TachographPayload”) is defined in 5.4.4/5.4.5 below and is updated at the interval determined in 4.1.1.1 (*DSC\_12*).
- DSRC-VU. This is the function, within or connected to the antenna and in communication with the VU through a wired or wireless (BLE) connection, which holds the current data (*VUPM-data*) and manages the response to an interrogation across the 5.8 GHz DSRC medium. Disconnection of the DSRC facility or interference during normal vehicle operation with the functioning of the DSRC facility shall be construed as a violation of Regulation (EU) N°. 165/2014.
- Security module (REDCR) (*SM-REDCR*) is the function used to decrypt and check integrity of the data originating from the VU. The means by which this is achieved is determined in Appendix 11 Common Security Mechanisms, and is not defined in this Appendix.
- The DSRC facility (REDCR) (*DSRC-REDCR*) function comprises a 5.8 GHz transceiver and associated firmware and software which manages *the Communication* with the *DSRC-VU* according to this Appendix.
- The *DSRC-REDCR* interrogates the *DSRC-VU* of the targeted vehicle and obtains *the Data* (the targeted vehicle’s current *VUPM-data*) via the DSRC link and processes and stores the received data in its *SM-REDCR*.
- The DSRC-VU antenna shall be positioned at a location where it optimizes the DSRC communication between the vehicle and the roadside antenna (in general in or close to the centre of the vehicle windshield ...). For light vehicles an installation corresponding to the upper part of the windscreen is suitable.
  - There shall be no metal objects (e.g. name badges, stickers, foil anti reflection (tinting) strips, sun visors, windshield wiper at rest) in front of, or close to the antenna, that can interfere with the communication.
  - The antenna shall be mounted so that its boresight approximately is parallel with the surface of the road.

DSC\_20 The Antenna and The Communication shall operate within ERC 70-03, tested against the appropriate parameters of EN 300 674-1 as described in section 4. The Antenna and the Communication can implement mitigation techniques against the risk of wireless interference as described in ECC report 228 using e.g., filters in the CEN DSRC 5.8 GHz communication.

DSC\_21 The DSRC antenna shall be connected to the DSRC-VU facility either directly within the module mounted to or close to the windshield, or through a dedicated cable constructed in a manner to make illegal disconnection difficult. Disconnection of or interference with the functioning of Antenna shall be a violation of Regulation (EU) N°. 165/2014. Deliberate masking or otherwise detrimentally affecting the operational performance of the Antenna shall be construed as a violation of Regulation (EU) N°. 165/2014.

DSC\_22 The form factor of the antenna is not defined and shall be a commercial decision, so long as the fitted DSRC-VU meets the conformance requirements defined in section 5 below. The antenna shall be positioned as determined in DSC\_19 and shown in figure 14.4 (oval line) and it efficiently supports the use cases described in in 3.1.2 and 3.1.3.





**Figure 14.4 — Example of positioning of the 5.8 GHz DSRC antenna in the windshield of regulated vehicles**

The form factor of *the REDCR* and its antenna may vary according to the circumstances of the reader (tripod mounted, hand held, vehicle mounted, etc.) and the *modus operandi* employed by the agent of the competent control authorities.

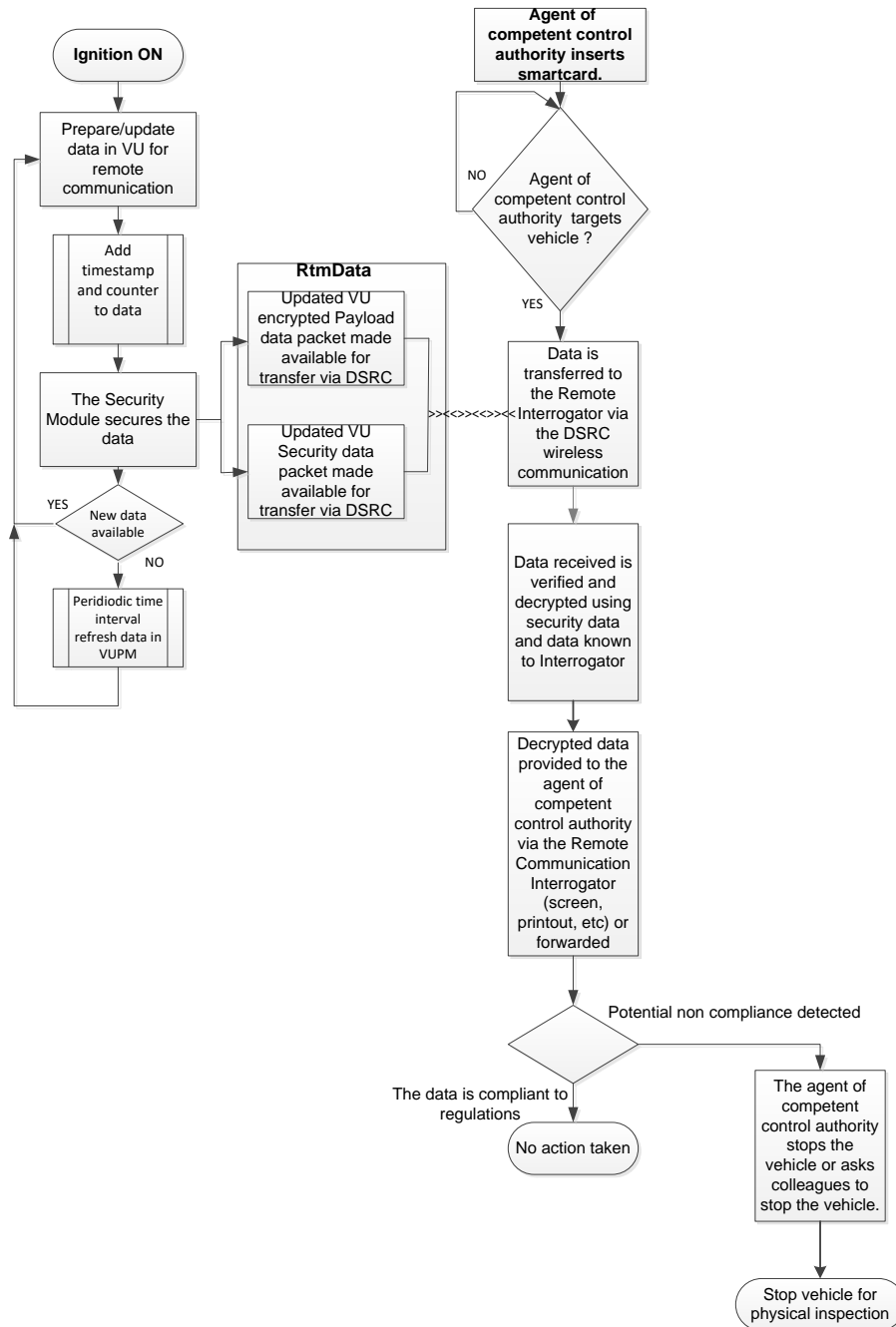
A display and/or notification function is used to present the results of the remote communication function to the agent of the competent control authorities. A display may be provided on a screen, as a printed output, an audio signal, or a combination of such notifications. The form of such display and/or notification is a matter of the requirements of the agents of the competent control authorities and equipment design and is not specified within this Appendix.

- DSC\_23 The design and form factor of the *REDCR* shall be a function of commercial design, operating within ERC 70-03, and the design and performance specifications defined in this Appendix, (section 5.3.2), thus providing the marketplace maximum flexibility to design and provide equipment to cover the specific interrogation scenarios of any particular competent control authority.
- DSC\_24 The design and form factor of the *DSRC-VU* and its positioning inside or outside the VU shall be a function of commercial design, operating within ERC 70-03 and the design and performance specifications defined in this Appendix (section 5.3.2) and within this Clause (5.1).
- DSC\_25 However, the *DSRC-VU* shall be reasonably capable to accept data concept values from other intelligent vehicle equipment by means of an open industry standard connection and protocols. (For example from weigh on board equipment), so long as such data concepts are identified by unique and known application identifiers/file names, and the instructions to operate such protocols shall be made available to the European Commission, and available without charge to manufacturers of relevant equipment.

4.2 Workflow

4.2.1 Operations

The workflow of operations is represented in Figure 14.5.



**Figure 14.5 — Workflow for remote communication function**

The steps are described below:

- a. Whenever the vehicle is in operation (ignition ON) the tachograph is providing data to the VU function. The VU function prepares *the Data* for the remote communication function (encrypted) and updates the *VUPM* held in the memory of the *DSRC-VU* (as defined in 4.1.1.1 - 4.1.1.2). *The Data* collected shall be formatted as determined in 5.4.4 – 5.4.5 below.
- b. On every occasion that *the Data* is updated, the timestamp defined in the security data concept shall be updated.
- c. The *VUSM* function secures the data in accordance with the procedures determined in Appendix 11.
- d. On every occasion that *the Data* is updated (see 4.1.1.1 - 4.1.1.2), *the Data* shall be transferred to the *DSRC-VU*, where it replaces any previous data, in order that updated current data (*the Data*) shall always be available to be provided in the event of an interrogation by an *REDCR*. When supplied by the VU to the *DSRC-VU* *the Data* shall be identifiable by the filename *RTMData* or by ApplicationID and Attribute identifiers.
- e. If an agent of the competent control authorities wishes to target a vehicle and collect *the Data* from the targeted vehicle, the agent of the competent control authorities shall first insert his/her smartcard in *the REDCR* to enable *the Communication* and to allow the *SM-REDCR* to verify its authenticity and decrypt the data.
- f. The agent of the competent control authority then targets a vehicle and requests the data through remote communication. *The REDCR* opens a 5.8 GHz DSRC interface session with the *DSRC-VU* of the targeted vehicle, and requests *the Data*. *The Data* is transferred to *the REDCR* through the wireless communication system as a DSRC Attribute using the Application service GET as defined in 5.4. The Attribute contains the encrypted payload data values and the DSRC security data.
- g. The data is analyzed by the *REDCR* equipment and provided to the agent of the competent control authority.
- h. The agent of the competent control authority uses the data to assist in a decision of whether or not to stop for a detailed inspection, or ask another agent of the competent control authority to stop the vehicle.

#### 4.2.2 Interpretation of the Data received via the DSRC communication

DSC\_26 Data received across the 5.8 GHz interface shall carry the meaning and import defined in 5.4.4 and 5.4.5 below and only that meaning and import, and shall be understood within the objectives defined therein. In accordance with the provisions of Regulation (EU) N°. 165/2014, *the Data* shall be used only to provide relevant information to a competent control authority to assist them to determine which vehicle should be stopped for physical inspection, and shall be subsequently destroyed in accordance with Article 9 of Regulation (EU) N°. 165/2014.

#### 4.3 DSRC Physical interface parameters for remote communication

##### 4.3.1 Location constraints

DSC\_27 The remote interrogation of vehicles using a 5.8GHz DSRC interface should not be used within 200 metres of an operational 5.8 GHz DSRC gantry.

##### 4.3.2 Downlink and uplink parameters

DSC\_28 The equipment used for remote tachograph monitoring shall conform to and operate within ERC70-03 and the parameters defined in Tables 14.1 and 14.2 below.

DSC\_29 Further, to ensure compatibility with the operational parameters of other standardised 5.8 GHz DSRC systems, the equipment used for remote tachograph monitoring shall conform to parameters from EN 12253 and EN 13372.

Namely:

**Table 14.1 — Downlink parameters**

<i>Item No.</i>	<i>Parameter</i>	<i>Value(s)</i>	<i>Remark</i>
D1	Downlink Carrier Frequencies	There are four alternatives which may be used by an REDCR : 5.7975 GHz 5.8025 GHz 5.8075 GHz	Within ERC 70-03. Carrier Frequencies may be selected by the implementer of the roadside system and need not be known in the DSRC-VU
D1a (*)	Tolerance of Carrier	within $\square$ 5 ppm	(Consistent with EN 12253)
D2(*)	RSU (REDCR) Transmitter Spectrum Mask	Within ERC 70-03. REDCR shall be according to Class B,C as defined in EN 12253 .	Parameter used for controlling interference between interrogators in proximity (as defined in EN 12253 and EN 13372).
		No other specific requirement within this Annex	
D3	OBU(DSRC-VU) Minimum Frequency	5.795 – 5.815 GHz	(Consistent with EN 12253)
D4 (*)	Maximum E.I.R.P.	Within ERC 70-03 (unlicensed) and within National Regulation	(Consistent with EN 12253)
D4a	Angular E.I.R.P. mask	According to declared and published specification of interrogator designer	(Consistent with EN 12253)
D5	Polarisation	Left hand circular	(Consistent with EN 12253)
D5a	Cross-Polarisation	XPD: In bore sight: (REDCR) RSU t $\square\square$ 15 dB  (DSRC-VU) OBU r $\square\square$ 10 dB	(Consistent with EN 12253)
D6 (*)	Modulation	Two level amplitude modulation.	(Consistent with EN 12253)

<i>Item No.</i>	<i>Parameter</i>	<i>Value(s)</i>	<i>Remark</i>
<b>D6a (*)</b>	Modulation Index	0.5 ... 0.9	(Consistent with EN 12253)
<b>D6b</b>	Eye Pattern	$\geq 90$ % (time) / $\geq 85$ % (amplitude)	
<b>D7 (*)</b>	Data Coding	FM0 "1" bit has transitions only at the beginning and end of the bit interval. "0" bit has an additional transition in the middle of the bit interval	(Consistent with EN 12253)
<b>D8 (*)</b>	Bit rate	500 kBit/s	(Consistent with EN 12253)
<b>D8a</b>	Tolerance of Bit Clock	better than $\square$ 100 ppm	(Consistent with EN 12253)
<b>D9(*)</b>	Bit Error Rate (B.E.R.)	$\leq 10^{-6}$ when incident power at OBU (DSRC-VU) is in the range given by [D11a to D11b].	(Consistent with EN 12253)
<b>D10</b>	Wake-up trigger for OBU (DSRC-VU)	OBU (DSRC-VU) shall wake up on receiving any frame with 11 or more octets (including preamble)	No special wake-up pattern is necessary.  DSRC-VU may wake up on receiving a frame with less than 11 octets
<b>D10a</b>	Maximum Start Time	$\leq 5$ ms	(Consistent with EN 12253)
<b>D11</b>	Communication zone	Spatial region within which a B.E.R. according to D9a is	(Consistent with EN 12253)
<b>D11a (*)</b>	Power Limit for communication	-24dBm	(Consistent with EN 12253)
<b>D11b (*)</b>	Power Limit for communication (lower).	Incident power: -43 dBm (boresight) -41 dBm (within $-45^\circ$ - $+45^\circ$ corresponding to the plane parallel to the road surface	(Consistent with EN 12253)  Extended requirement for horizontal angles up to
<b>D12(*)</b>	Cut-off power level of DSRC VU	-60 dBm	(Consistent with EN 12253)
<b>D13</b>	Preamble	Preamble is mandatory.	(Consistent with EN 12253)

<i>Item No.</i>	<i>Parameter</i>	<i>Value(s)</i>	<i>Remark</i>
<b>D13a</b>	Preamble Length and	16 bits □ 1 bit of FM0 coded "1" bits	(Consistent with EN 12253)
<b>D13b</b>	Preamble Wave form	An alternating sequence of low level and high level with pulse duration of  2 μs.	(Consistent with EN 12253)
<b>D13c</b>	Trailing Bits	The RSU (REDCR) is permitted to transmit a maximum of 8 bits after the end flag. An OBU (DSRC-VU) is not required to take these additional bits into	(Consistent with EN 12253)

(\*) - Downlink parameters subject to conformance testing in accordance with relevant parameter test from EN 300 674-1

**Table 14.2 — Uplink parameters**

<i>Item No.</i>	<i>Parameter</i>	<i>Value(s)</i>	<i>Remark</i>
<b>U1 (*)</b>	Sub-carrier Frequencies	A OBU (DSRC-VU) shall support 1.5 MHz and  2.0 MHz	Selection of sub-carrier frequency  (1.5 MHz or 2.0 MHz) depends on the EN 13372 profile selected.
<b>U1a(*)</b>	Tolerance of Sub-carrier Frequencies	An RSU (REDCR) shall support 1.5 MHz or within □ 0.1%	(Consistent with EN 12253)
<b>U1b</b>	Use of Side Bands	Same data on both sides	(Consistent with EN 12253)
<b>U2 (*)</b>	OBU (DSRC-VU) Transmitter Spectrum Mask	According to EN12253 1) Out band power: see ETSI EN 300674-1 2) In band power: [U4a] dBm in 500 kHz 3) Emission in any other uplink	(Consistent with EN 12253)
<b>U4a (*)</b>	Maximum Single Side Band E.I.R.P. (boresight)	Two options: U4a-0: -14 dBm U4a-1: -21 dBm	According to declared and published specification of equipment designer

<i>Item No.</i>	<i>Parameter</i>	<i>Value(s)</i>	<i>Remark</i>
<b>U4b (*)</b>	Maximum Single Side Band E.I.R.P. (35 <sup>0</sup> )	Two options: - Not applicable - -17dBm	According to declared and published specification of equipment designer
<b>U5</b>	Polarisation	Left hand circular	(Consistent with EN 12253)
<b>U5a</b>	Cross Polarisation	XPB: In bore sight: (REDCR) RSU r □□15 dB (DSRC-VU) OBU t □□10 dB	(Consistent with EN 12253)
<b>U6</b>	Sub-Carrier Modulation	2-PSK	(Consistent with EN 12253)
<b>U6b</b>	Duty Cycle	Encoded data synchronised with sub-carrier: Transitions of encoded data coincide with Duty Cycle: 50% ± α, α ≤ 5%	(Consistent with EN 12253)
<b>U6c</b>	Modulation on Carrier	Multiplication of modulated sub- carrier with carrier.	(Consistent with EN 12253)
<b>U7 (*)</b>	Data Coding	NRZI (No transition at beginning of "1" bit, transition at beginning of "0")	(Consistent with EN 12253)
<b>U8 (*)</b>	Bit Rate	250 kbit/s	(Consistent with EN 12253)
<b>U8a</b>	Tolerance of Bit Clock	Within □□1000 ppm	(Consistent with EN 12253)
<b>U9</b>	Bit Error Rate (B.E.R.) for Communication	≤10 <sup>-6</sup>	(Consistent with EN 12253)
<b>U11</b>	Zone	The spatial region within which the DSRC-VU is situated such that its transmissions are received by the REDCR with a B.E.R. of less than that given by U9a.	(Consistent with EN 12253)
<b>U12a(*)</b>	Conversion Gain (lower limit)	1 dB for each side band Range of angle: Circularly symmetric between bore sight and ± 35° and within -45° - +45°	Greater than the specified value range for horizontal angles up to ±45°, due to



<i>Item No.</i>	<i>Parameter</i>	<i>Value(s)</i>	<i>Remark</i>
<b>U12b(*)</b>	Conversion Gain (upper limit)	10 dB for each side band	Less than the specified value range for each side band within a circular cone around boresight of  <input type="checkbox"/> 45° opening angle
<b>U13</b>	Preamble	Preamble is mandatory.	(Consistent with EN 12253)
<b>U13a</b>	Preamble Length and Pattern	32 to 36 $\mu$ s modulated with sub-carrier only, then 8 bits of NRZI coded "0" bits.	(Consistent with EN 12253)
<b>U13b</b>	Trailing Bits	The DSRC-VU is permitted to transmit a maximum of 8 bits after the end flag. A RSU (REDCR) is not required to take these additional bits into	(Consistent with EN 12253)

(\*) - Uplink parameters subject to conformance testing in accordance with relevant parameter test from EN 300 674-1

### 4.3.3 Antenna design

#### 4.3.3.1 REDCR antenna

DSC\_30 The design of the *REDCR* antenna shall be a function of commercial design, operating within the limits defined in 5.3.2 which is adapted to optimise the reading performance of the *DSRC-REDCR* for the specific purpose and read circumstances in which the *REDCR* has been designed to operate.

#### 4.3.3.2 VU antenna

DSC\_31 The design of the *DSRC-VU* antenna shall be a function of commercial design, operating within the limits defined in 5.3.2 which is adapted to optimise the reading performance of the *DSRC-REDCR* for the specific purpose and read circumstances in which the *REDCR* has been designed to operate.

DSC\_32 The VU antenna shall be fixed to, or close to, the front windshield of the vehicle as specified in 5.1 above.

DSC\_33 In the test environment in a workshop (see section 6.3), a DSRC-VU antenna, affixed according to 5.1 above, shall successfully connect with a standard test communication and successfully provide an RTM transaction as defined within this Appendix, at a distance between 2 and 10 meters, better than 99% of the time, averaged over 1000 read interrogations.

## 4.4 DSRC Protocol requirements for RTM

### 4.4.1 Overview

DSC\_34 The transaction protocol to download *the Data* across the 5.8 GHz DSRC interface link shall be according to the following steps. This section describes a transaction flow under ideal conditions without retransmissions or communication interrupts.

NOTE The purpose of the initialisation phase (Step 1) is to set up the communication between the *REDCR* and DSRC-VUs that have entered the 5.8 GHz DSRC (master-slave) transaction zone but have not yet established communication with the *REDCR*, and to notify the application processes.

↳ **Step 1** Initialisation. The *REDCR* sends a frame containing a 'beacon service table' (BST) that includes the application identifiers (AIDs) in the service list that it supports. In the RTM

application this will simply be the service with the AID value = 2 (Freight&Fleet). The *DSRC-VU* evaluates the received BST, and shall respond (see below) with the list of the supported applications within the Freight&Fleet domain, or shall not respond if none are supported. If the *REDCR* does not offer AID=2, the *DSRC-VU* shall not answer to the *REDCR*.

- ↳ **Step 2** The *DSRC-VU* sends a frame containing a request for a private window allocation.
- ↳ **Step 3** The *REDCR* sends a frame containing a private window allocation.
- ↳ **Step 4** The *DSRC-VU* uses the allocated private window to send a frame containing its vehicle service table (VST). This VST includes a list of all the different application instantiations that this *DSRC-VU* supports in the framework of AID=2. The different instantiations shall be identified by means of uniquely generated EIDs, each associated with an Application Context Mark parameter value indicating the application and standard supported.
- ↳ **Step 5** Next the *REDCR* analyses the offered VST, and either terminates the connection (RELEASE) since it is not interested in anything the VST has to offer (i.e. it is receiving a VST from a *DSRC-VU* that is not supporting the RTM transaction), or, if it receives an appropriate VST it starts an app instantiation.
- ↳ **Step 6** To bring this about, the *REDCR* shall send a frame containing a command to retrieve the RTM data, identifying the RTM application instantiation by specifying the identifier corresponding to the RTM application instantiation (as specified by the *DSRC-VU* in the VST), and shall allocate a private window.
- ↳ **Step 7** The *DSRC-VU* uses the newly allocated private window to send a frame that contains the addressed identifier corresponding to the RTM application instantiation as provided in the VST, followed by the attribute *RtmData* (payload element + security element).
- ↳ **Step 8** If there are multiple services requested, the value 'n' is changed to the next service reference number and the process repeated.
- ↳ **Step 9** The *REDCR* confirms receipt of the data by sending a frame containing a RELEASE command to the *DSRC-VU* to terminate the session OR if it has failed to validate a successful receipt of the LDPU goes back to step 6.

See Figure 14.6 for a pictorial description of the transaction protocol.

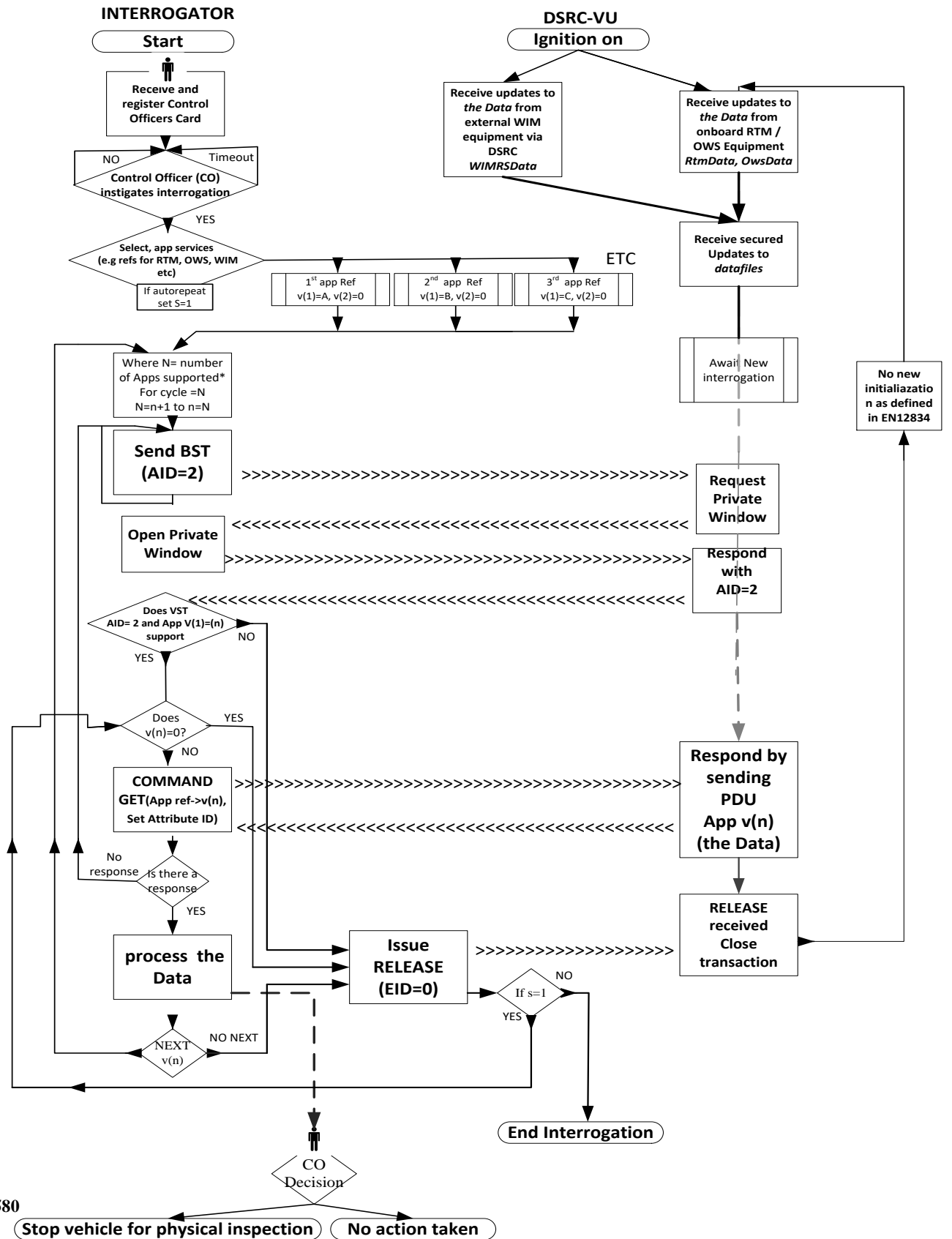


Figure 14.6 — RTM over 5.8 GHz DSRC process flow

4.4.2 Commands

DSC\_35 The following commands are the only functions used in an RTM transaction phase

- **INITIALISATION.request:** A command, issued from the REDCR in the form of a broadcast with definition of applications that the REDCR supports.
- **INITIALISATION.response:** An answer from the DSRC-VU confirming the connection and containing a list of supported application instances with characteristics and information how to address them (EID).
- **GET.request:** A command, issued from *the REDCR* to the *DSRC-VU*, that specifies the application instantiation to be addressed by means of a defined EID, as received in the VST, instructing the *DSRC-VU* to send the selected attribute(s) with *the Data*. The objective of the GET command is for *the REDCR* to obtain *the Data* from the *DSRC-VU*.
- **GET.response:** An answer from the DSRC-VU that contains *the Data* requested.
- **ACTION.request ECHO:** A command, instructing the *DSRC-VU* to send back data from the *DSRC-VU* to *the REDCR*. The objective of the ECHO command is to enable workshops or type approval test facilities to test that the DSRC link is working without needing access to security credentials.
- **ACTION.response ECHO:** An answer from the DSRC VU on the ECHO command.
- **EVENT\_REPORT.request RELEASE:** A command, instructing the *DSRC-VU* that the transaction is ended. The objective of the RELEASE command is to end the session with the *DSRC-VU*. On receipt of the RELEASE the *DSRC-VU* shall not respond to any further interrogations under the current connection. Note that according to EN 12834 a DSRC-VU will not connect twice to the same interrogator unless it has been out of the communication zone for 255 seconds or if the Beacon ID of the interrogator is changed.

4.4.3 Interrogation command sequence

DSC\_36 From the perspective of the command and response sequence, the transaction is described as follows:

Sequence	Sender		Receiver	Description	A
1	REDCR	>	DSRC-VU	Initialisation of the communication link – Request	REDCR broadcasts BST
2	DSRC-VU	>	REDCR	Initialisation of the communication link – Response	If BST supports AID=2 then DSRC-VU Requests a private window
3	REDCR	>	DSRC-VU	Grants a private window	Sends Frame containing private
4	DSRC-VU	>	REDCR	Sends VST	Sends Frame
5	REDCR	>	DSRC-VU	Sends GET.request for data in Attribute for specific EID	
6	DSRC-VU	>	REDCR	Sends GET.response with requested Attribute for specific EID	Sends Attribute (RTMData, OWSDData....

7	REDCR	>	DSRC-VU	Sends GET.request for data other Attribute (if appropriate)”	
8	DSRC-VU	>	REDCR	Sends GET.response with requested Attribute	Sends Attribute with data for specific EID
9	REDCR	>	DSRC-VU	Acknowledges successful receipt of	Sends RELEASE command which closes
10	DSRC-VU			Closes transaction	

An example of the transaction sequence and contents of the exchanged frames is defined in clauses 5.4.7 and 5.4.8

#### 4.4.4 Data structures

DSC\_37 The semantic structure of *the Data* when passed across the 5.8 GHz DSRC interface shall be consistent with what described in this Appendix. The way these data are structured is specified in this clause.

DSC\_38 The payload (RTM data) consists of the concatenation of

1. EncryptedTachographPayload data, which is the encryption of the TachographPayload defined in ASN.1 in section 5.4.5. The method of encryption is described in Appendix 11
2. DSRCSecurityData, specified in Appendix 11.

DSC\_39 The RTM Data is being addressed as RTM Attribute=1 and is transferred in the RTM container =10.

DSC\_40 The RTM Context Mark shall identify the supported standard part in the TARV series of standards (RTM corresponds to Part 9)

4.4.5 The ASN.1 module definition for the DSRC data within the RTM application is defined as follows:

4.4.6

4.4.7 TarvRtm {iso(1) standard(0) 15638 part9(9) version1(1)} DEFINITIONS AUTOMATIC TAGS

4.4.8 ::= BEGIN

4.4.9 IMPORTS

4.4.10 -- Imports data attributes and elements from EFC which are used for RTM

4.4.11 LPN

4.4.12 FROM EfcDsrcApplication {iso(1) standard(0) 14906 application(0) version5(5)}

4.4.13

4.4.14 -- Imports function parameters from the EFC Application Interface Definition

4.4.15 SetMMIRq

4.4.16 FROM EfcDsrcApplication {iso(1) standard(0) 14906 application(0) version5(5)}

4.4.17

4.4.18 -- Imports the L7 DSRCData module data from the EFC Application Interface Definition

4.4.19 Action-Request, Action-Response, ActionType, ApplicationList, AttributeIdList, AttributeList, Attributes,

4.4.20 BeaconID, BST, Dsrc-EID, DSRCApplicationEntityID, Event-Report-Request, Event-Report-Response,

4.4.21 EventType, Get-Request, Get-Response, Initialisation-Request, Initialisation-Response,

4.4.22 ObeConfiguration, Profile, ReturnStatus, Time, T-APDUs, VST

4.4.23 FROM EfcDsrcGeneric {iso(1) standard(0) 14906 generic(1) version5(5)};

4.4.24

4.4.25

4.4.26 -- Definitions of the RTM functions:

4.4.27 RTM-InitialiseComm-Request ::= BST

4.4.28 RTM-InitialiseComm-Response ::= VST

4.4.29 RTM-DataRetrieval-Request ::= Get-Request (WITH COMPONENTS {fill (SIZE(1)), eid, accessCredentials ABSENT, iid ABSENT, attrIdList})

- 4.4.30 RTM-DataRetrieval-Response ::= Get-Response {RtmContainer} (WITH COMPONENTS {..., eid, iid ABSENT})
- 4.4.31 RTM-TerminateComm ::= Event-Report-Request {RtmContainer} (WITH COMPONENTS {mode (FALSE), eid (0),
- 4.4.32 eventType (0)})
- 4.4.33
- 4.4.34 RTM-TestComm-Request ::= Action-Request {RtmContainer} (WITH COMPONENTS {..., eid (0), actionType
- 4.4.35 (15), accessCredentials ABSENT, iid ABSENT})
- 4.4.36
- 4.4.37 RTM-TestComm-Response ::= Action-Response {RtmContainer} (WITH COMPONENTS {..., fill (SIZE(1)), eid
- 4.4.38 (0), iid ABSENT})
- 4.4.39
- 4.4.40 -- Definitions of the RTM attributes:
- 4.4.41 RtmData ::= SEQUENCE {
- 4.4.42     encryptedTachographPayload OCTET STRING (SIZE(67)) (CONSTRAINED BY { -- calculated encrypting TachographPayload as per Appendix 11 --}),
- 4.4.43     DSRCSecurityData OCTET STRING
- 4.4.44     }
- 4.4.45 TachographPayload ::= SEQUENCE {
- 4.4.46     tp15638VehicleRegistrationPlate LPN -- Vehicle Registration Plate as per EN 15509.
- 4.4.47     tp15638SpeedingEvent     BOOLEAN, -- 1= Irregularities in speed (see Annex 1C)
- 4.4.48     tp15638DrivingWithoutValidCard BOOLEAN, -- 1= Invalid card usage (see Annex 1C)
- 4.4.49     tp15638DriverCard     BOOLEAN, -- 0= Indicates a valid driver card (see Annex 1C)
- 4.4.50     tp15638CardInsertion     BOOLEAN, -- 1= Card insertion while driving (see Annex 1C)
- 4.4.51     tp15638MotionDataError    BOOLEAN, -- 1= Motion data error (see Annex 1C)
- 4.4.52     tp15638VehicleMotionConflict    BOOLEAN, -- 1= Motion conflict (see Annex 1C)



- 
- 4.4.53 tp156382ndDriverCard BOOLEAN, -- 1= Second driver card inserted (see Annex 1C)
- 4.4.54 tp15638CurrentActivityDriving BOOLEAN, -- 1= other activity selected;
- 4.4.55 -- 0= driving selected
- 4.4.56 tp15638LastSessionClosed BOOLEAN, -- 1= improperly, 0= properly, closed
- 4.4.57 tp15638PowerSupplyInterruption INTEGER (0..127), -- Supply interrupts in the last 10 days
- 4.4.58 tp15638SensorFault INTEGER (0..255),-- eventFaultType as per data dictionary
- 4.4.59 -- All subsequent time related types as defined in Annex 1C.
- 4.4.60 tp15638TimeAdjustment INTEGER(0..4294967295), -- Time of the last time adjustment
- 4.4.61 tp15638LatestBreachAttempt INTEGER(0..4294967295), -- Time of last breach attempt
- 4.4.62 tp15638LastCalibrationData INTEGER(0..4294967295), -- Time of last calibration data
- 4.4.63 tp15638PrevCalibrationData INTEGER(0..4294967295), -- Time of previous calibration data
- 4.4.64 tp15638DateTachoConnected INTEGER(0..4294967295), -- Date tachograph connected
- 4.4.65 tp15638CurrentSpeed INTEGER (0..255), -- Last current recorded speed
- 4.4.66 tp15638Timestamp INTEGER(0..4294967295) -- Timestamp of current record2
- 4.4.67 }
- 4.4.68 Rtm-ContextMark ::= SEQUENCE {
- 4.4.69 standardIdentifier StandardIdentifier, -- identifier of the TARV part and its version
- 4.4.70
- 4.4.71 RtmCommProfile INTEGER {
- 4.4.72 C1 (1),
- 4.4.73 C2 (2)
- 4.4.74 } (0..255) DEFAULT 1
- 4.4.75 }
- 4.4.76 RtmTransferAck ::= INTEGER {
- 4.4.77 Ok (1),

4.4.78                    NoK (2)  
4.4.79                    } SIZE (1..255)  
4.4.80  
4.4.81   StandardIdentifier ::= OBJECT IDENTIFIER  
4.4.82   RtmContainer ::= CHOICE {  
4.4.83       integer            [0] INTEGER,  
4.4.84       bitstring         [1] BIT STRING,  
4.4.85       octetstring        [2] OCTET STRING (SIZE (0..127, ...)),  
4.4.86       universalString    [3] UniversalString,  
4.4.87       beaconId          [4] BeaconID,  
4.4.88       t-apdu             [5] T-APDUs,  
4.4.89       dsrcApplicationEntityId [6] DSRCApplicationEntityID,  
4.4.90       dsrc-Ase-Id        [7] Dsrc-EID,  
4.4.91       attrIdList         [8] AttributeIdList,  
4.4.92       attrList            [9] AttributeList{RtmContainer},  
4.4.93       rtmData            [10] RtmData,  
4.4.94       rtmContextmark      [11] Rtm-ContextMark,  
4.4.95       reserved12         [12] NULL,  
4.4.96       reserved13         [13] NULL,  
4.4.97       reserved14         [14] NULL,  
4.4.98       time                [15] Time,  
4.4.99       -- values from 16 to 255 reserved for ISO/CEN usage  
4.4.100       } }

END

## 4.4.101 Elements of RtmData, actions performed and definitions

DSC\_41 The data values to be calculated by the VU and used to update the secured data in the DSRC-VU shall be calculated according to the rules defined in Table 14.3:

**Table 14.3 — Elements of RtmData, actions performed and definitions**

(1)	(2)	(3)
<i>RTM Data Element</i>	<i>Action performed by the VU</i>	<i>ASN.1 definition of data</i>
<b>RTM1 Vehicle Registration Plate</b>	The VU shall set the value of the <i>tp15638VehicleRegistrationPlate</i> data element RTM1 from the recorded value of the data type <i>VehicleRegistrationIdentification</i> as defined in Appendix 1 <i>VehicleRegistrationIdentification</i>	Vehicle Registration Plate expressed as a string of characters  14906 with the limitation specified in EN 15509 which is a SEQUENCE comprising Country Code followed by an alphabet indicator followed by the plate number itself, which is always 14 octets (padded with <i>tp15638speedingEvent</i> BOOLEAN,
<b>RTM2 Speeding Event</b>	The VU shall generate a boolean value for data element RTM2 <i>tp15638SpeedingEvent</i> .  The <i>tp15638SpeedingEvent</i> value shall be calculated by the VU from the number of Over Speeding Events recorded in the VU in the last 10 days of occurrence, as defined in Annex 1C.	1 (TRUE) - Indicates irregularities in speed within last 10 days of occurrence

(1) RTM Data Element	(2) Action performed by the VU	(3) ASN.1 definition of data
<b>RTM3 Driving Without Valid Card</b>	<p>The VU shall generate a boolean value for data element RTM3</p> <p>tp15638DrivingWithoutValidCard.</p>	<p>1 (TRUE) = Indicates invalid card usage</p>
<b>RTM4 Valid Driver Card</b>	<p>The VU shall assign a value of True to the tp15638DrivingWithoutValidCard variable if the VU data has recorded at least one event in the last 10 days of occurrence of</p> <p>The VU shall generate a boolean value for data element RTM4</p> <p>tp15638DriverCard on the basis of the data stored in the VU and defined in Appendix 1.</p>	<p>0 (FALSE) = Indicates a valid driver card</p>
<b>RTM5 Card Insertion while Driving</b>	<p>The VU shall generate a boolean value for data element RTM5.</p> <p>The VU shall assign a value of TRUE to the tp15638CardInsertion variable if the VU data has recorded in the last 10 days of occurrence at least one event of type “Card insertion while driving.” as defined in</p>	<p>1 (TRUE) = Indicates card insertion while driving within last 10 days of occurrence</p>

(1) RTM Data Element	(2) Action performed by the VU		(3) ASN.1 definition of data
<b>RTM6</b> <b>Motion Data Error</b>	The VU shall generate a boolean value for data element RTM6.	1 (TRUE) = Indicates motion data error within last 10 days of occurrence	tp15638motionDataErr or BOOLEAN,
<b>RTM7</b> <b>Vehicle Motion Conflict</b>	The VU shall assign a value of TRUE to the tp15638MotionDataError variable if the VU data has in the last 10 days of occurrence recorded at least one event of tvne The VU shall generate a boolean value for data element RTM7.	1 (TRUE) = Indicates motion conflict within last 10 days of occurrence	tp15638vehicleMotionConflict BOOLEAN,
<b>RTM8</b> <b>2nd Driver Card</b>	The VU shall assign a value of TRUE to the tp15638vehicleMotionConflict variable if the VU data has in the last 10 days recorded at least The VU shall generate a boolean value for data element RTM8 on the basis of Annex 1C (“Driver Activity Data” CREW and CO-DRIVER).	1 (TRUE) = Indicates a second driver card inserted	tp156382ndDriverCard BOOLEAN,
<b>RTM9</b> <b>Current Activity</b>	If a 2nd valid driver card The VU shall generate a boolean value for data element RTM9.  If the current activity is recorded in the VU as any activity other than “DRIVING” as defined	1 (TRUE) = other activity selected; 0 (FALSE) = driving selected	tp15638currentActivityDriving BOOLEAN

(1) RTM Data Element	(2) Action performed by the VU	(3) ASN.1 definition of data
<b>RTM10 Last Session Closed</b>	The VU shall generate a boolean value for data element RTM10.  If the last card session was not properly closed as defined in Annex 1C the VU shall set the variable to TRUE.	1 (TRUE) = improperly closed 0 (FALSE) = properly closed
<b>RTM11 Power Supply Interruption</b>	The VU shall generate an integer value for data element RTM11.  The VU shall assign a value for the tp15638PowerSupplyInterruption variable equal to the longest power supply interruption” according to Article 9, Reg (EU)	-- Number of power supply interruptions in last 10 days of occurrence
		tp15638lastSessionClosed  BOOLEAN  tp15638powerSupplyInterruption INTEGER (0..127),

(1)	(2)	(3)
<i>RTM Data Element</i>	<i>Action performed by the VU</i>	<i>ASN.1 definition of data</i>
<b>RTM12 Sensor Fault</b>	The VU shall generate an integer value for data element RTM12.	--sensor fault one octet as per data dictionary

The VU shall assign to the variable sensorFault a value of:

- 1 if an event of type '35'H Sensor fault has been recorded in the last 10 days,
- 2 if an event of type GNSS receiver fault (either internal or external with enum values '51'H or '52'H) has been recorded in the last 10 days.
- 3 if an event of type '53'H External GNSS communication fault has been recorded in the last 10 days of occurrence.

(1) <i>RTM Data Element</i>	(2) <i>Action performed by the VU</i>	(3) <i>ASN.1 definition of data</i>
<b>RTM13 Time Adjustment</b>	The VU shall generate an integer value (timeReal from Appendix 1) for data element RTM13 on the basis of the presence of Time Adjustment data as defined in Annex 1C.	Time of the last time adjustment  tp15638TimeAdjustment  INTEGER(0..4294967295),
<b>RTM14 Security Breach Attempt</b>	The VU shall assign the value of time at which the last time adjustment data  The VU shall generate an integer value (timeReal from Appendix 1) for data element RTM14 on the basis of the presence of a Security breach attempt event as defined in Annex 1C.	Time of last breach attempt  -- Default value =0x00FF  tp15638LatestBreachAttempt  INTEGER(0..4294967295),
<b>RTM15 Last Calibration</b>	The VU shall set the value of the time of the latest security breach  The VU shall generate an integer value (timeReal from Appendix 1) for data element RTM15 on the basis of the presence of Last Calibration data as defined in Annex 1C.	Time of last calibration data  tp15638LastCalibrationData  INTEGER(0..4294967295),
<b>RTM16 Previous Calibration</b>	The VU shall set the value of time of the latest two calibrations (RTM15 and RTM16) which are  The VU shall generate an integer value (timeReal from Appendix 1) for data element RTM16 of the calibration record preceding that of the last calibration  ELSE if there has been	Time of previous calibration data  tp15638PrevCalibrationData  INTEGER(0..4294967295),



(1) <i>RTM Data Element</i>	(2) <i>Action performed by the VU</i>	(3) <i>ASN.1 definition of data</i>
<b>RTM17 Date Tachograph Connected</b>	For data element RTM17 the VU shall generate an integer value (timeReal from Appendix 1).  The VU shall set the value of the time of the initial installation of the	Date tachograph connected  tp15638DateTachoConnected INTEGER(0..4294967295),
<b>RTM18 Current Speed</b>	The VU shall generate an integer value for data element RTM18.	Last current recorded speed  tp15638CurrentSpeed INTEGER (0..255),
<b>RTM19 Timestamp</b>	The VU shall set the value for RTM16 to the For data element RTM19 the VU shall generate an integer value (timeReal from Appendix 1).	Timestamp of current TachographPayload record  tp15638Timestamp INTEGER(0..4294967295),

<i>Octet #</i>	<i>Attribute/Field</i>	<i>Bits in octet</i>	<i>Description</i>
1	FLAG	0111 1110	Start flag
2	Broadcast ID	1111 1111	Broadcast address
3	MAC Control Field	1010 0000	Command PDU
4	LLC Control field	0000 0011	UI command
5	Fragmentation header	1xxx x001	No fragmentation
6	BST	1000	Initialisation request
	SEQUENCE {		
	OPTION indicator	0	NonMand applications not present
	BeaconID SEQUENCE {		
	ManufacturerId	INTEGER (0..65535)	
		xxx	Manufacturer Identifier
7		xxxx xxxx	

Octet#	Attribute/Field	Bits in octet	Description
8	IndividualID	xxxx x	27 bit ID available for manufacturer
9		xxxx xxxx	
10		xxxx xxxx	
11		xxxx xxxx	
12	Time	xxxx xxxx	32 bit UNIX real time
13		xxxx xxxx	
14		xxxx xxxx	
15		xxxx xxxx	
16	Profile	0000 0000	No extension. Example
17	MandApplications (SIZE(0..127,...)) OF {	0000 0001	No extension, Number of mandApplications = 1
18	SEQUENCE {		
	OPTION indicator	0	FID not present
	OPTION indicator	0	Parameter not present
	AID DSRCApplicationEntityID } }	00 0010	No extension. AID= 2 Freight&Fleet
19	ProfileList OF Profile }	0000 0000	No extension, number of profiles in
20	FCS	xxxx xxxx	Frame check sequence
21		xxxx xxxx	
22	Flag	0111 1110	End Flag

4.4.102 Data transfer mechanism

DSC\_42 Payload data defined previously are requested by the REDCR after initialisation phase, and consequently transmitted by the *DSRC-VU* in the allocated window. The command GET is used by the REDCR to retrieve data.

DSC\_43 For all DSRC exchanges, data shall be encoded using PER (Packed Encoding Rules).

4.4.103 Detailed DSRC transaction description

DSC\_44 Initialisation is performed according to DSC\_44 – DSC\_48 and Tables 14.4 – 14.9. In the initialisation phase, the REDCR starts sending a frame containing a BST (Beacon Service Table) according to EN 12834 and EN 13372, 6.2, 6.3, 6.4 and 7.1 with settings as specified in the following Table 14.4.

**Table 14.4 — Initialisation - BST frame settings**

Field	Settings
-------	----------

<i>Field</i>	<i>Settings</i>
Link Identifier	Broadcast address
BeaconId	As per EN 12834
Time	As per EN 12834
Profile	No extension, 0 or 1 to be used
MandApplications	No extension, EID not present, Parameter not present, AID= 2
NonMandApplications	<sup>Freight &amp; Fleet</sup> Not present
ProfileList	No extension, number of profiles in list = 0
Fragmentation header	No fragmentation
Layer 2 settings	Command PDU, UI command

A practical example of the settings specified in Table 14.4, with an indication of bit encodings, is given in the following Table 14.5.

**Table 14.5 — Initialisation - BST frame contents example**

DSC\_45 A DSRC-VU, when receiving a BST, requires the allocation of a private window, as specified by EN 12795 and EN 13372, 7.1.1, with no specific RTM settings. Table 14.6 provides an example of bit encoding.

**Table 14.6 — Initialisation - Private window allocation request frame contents**

<i>Octet #</i>	<i>Attribute/Field</i>	<i>Bits in octet</i>	<i>Description</i>
1	FLAG	0111 1110	Start flag
2	Private LID	xxxx xxxx	Link address of specific DSRC-VU
3		xxxx xxxx	
4		xxxx xxxx	
5		xxxx xxxx	
6	MAC Control field	0110 0000	Private window request
7	FCS	xxxx xxxx	Frame check sequence
8		xxxx xxxx	
9	Flag	0111 1110	End Flag

DSC\_46 The REDCR then answers by allocating a private window, as specified by EN 12795 and EN 13372, 7.1.1 with no specific RTM settings.

Table 14.7 provides an example of bit encoding.

**Table 14.7 — Initialisation - Private window allocation frame contents**

<i>Octet #</i>	<i>Attribute/Field</i>	<i>Bits in octet</i>	<i>Description</i>
----------------	------------------------	----------------------	--------------------

<i>Octet #</i>	<i>Attribute/Field</i>	<i>Bits in octet</i>	<i>Description</i>
1	FLAG	0111 1110	Start flag
2	Private LID	xxxx xxxx	Link address of the specific DSRC-VU
3		xxxx xxxx	
4		xxxx xxxx	
5		xxxx xxxx	
6	MAC Control field	0010 s000	Private window allocation
7	FCS	xxxx xxxx	Frame check sequence
8		xxxx xxxx	
9	Flag	0111 1110	End Flag

DSC\_47 The *DSRC-VU*, when receiving the private window allocation, sends its VST (Vehicle Service Table) as defined in EN 12834 and EN 13372, 6.2, 6.3, 6.4 and 7.1 with settings as specified Table 14.8, using the allocated transmission window.

**Table 14.8 — Initialisation - VST frame settings**

<i>Field</i>	<i>Settings</i>
Private LID	As per EN 12834
VST parameters	Fill=0, then for each supported application: EID present, parameter present,
Parameter	No extension, Contains the RTM Context Mark
ObeConfiguration	The optional ObeStatus field may be present, but shall not be used by the REDCR
Fragmentation	No fragmentation
Layer 2 settings	Command PDU, UI command

DSC\_48 The *DSRC-VU* shall support the “Freight and Fleet” application, identified by the Application Identifier ‘2’. Other Application Identifiers may be supported, but shall not be present in this VST, as the BST only requires AID=2. The “Applications” field contains a list of the supported application instances in the *DSRC-VU*. For each supported application instantiation, a reference to the appropriate standard is given, made of an Rtm Context mark, which is composed of an OBJECT IDENTIFIER representing the related standard, its part (9 for RTM) and possibly its version, plus an EID that is generated by the *DSRC-VU*, and associated to that application instance.

A practical example of the settings specified in Table 14.8, with an indication of bit encodings, is given in Table 14.9.

**Table 14.9 — Initialisation - VST frame contents example**

<i>Octet #</i>	<i>Attribute/Field</i>	<i>Bits in octet</i>	<i>Description</i>
1	FLAG	0111 1110	Start flag
2	Private LID	xxxx xxxx	Link address of the specific
3		xxxx xxxx	DSRC-VU
4		xxxx xxxx	
5		xxxx xxxx	
6	MAC Control field	1100 0000	Command PDU
7	LLC Control field	0000 0011	UI command
8	Fragmentation header	1xxx x001	No fragmentation
9	VST	1001	Initialisation response
	SEQUENCE {		
		0000	Unused and set to 0
10	Profile           INTEGER	0000 0000	No extension. Example profile 0
11		0000 0001	No extension. 1 application
12	SEQUENCE {		
	OPTION indicator	1	EID present
	OPTION indicator	1	Parameter present
		00 0010	No extension. AID= 2
13	EID           Dsrc-EID	xxxx xxxx	Defined within the OBU and identifying the application instance.
14	Parameter Container {	0000 0010	No extension, Container Choice = 02, Octet string
15		0000 1000	No extension, Rtm Context Mark length = 8
16	Rtm-ContextMark ::= SEQUENCE	0000 0110	Object Identifier of the supported
17	standardIdentifier	0000 0110	standard, part, and version.
18		0010 1000	Example: ISO (1) Standard (0)
19		1000 0000	TARV (15638) part9(9) Version1
20		1111 1010	(1).
21		0001 0110	First octet is 06H, which is the
22		0000 1001	Object Identifier Second octet is
23		0000 0001	06H, which is its length. Subsequent 6 octets encode the example Object Identifier
24	ObeConfiguration Sequence {		
	OPTION indicator	0	ObeStatus not present
	EquipmentClass    INTEGER	xxx xxxx	

<i>Octet #</i>	<i>Attribute/Field</i>	<i>Bits in octet</i>	<i>Description</i>
25	(0..32767)	xxxx xxxx	
26	ManufacturerId (0..65535)	INTEGER xxxx xxxx	Manufacturer identifier for the DSRC-VU as described in ISO 14816 Register
27		xxxx xxxx	
28	FCS	xxxx xxxx	Frame check sequence
29		xxxx xxxx	
30	Flag	0111 1110	End Flag

DCS\_49 The REDCR then reads the data by issuing a GET command, conforming to the GET command defined in EN 13372, 6.2, 6.3, 6.4 and EN 12834, with settings as specified in Table 14.10.

**Table 14.10 — Presentation - GET request frame settings**

<i>Field</i>	<i>Settings</i>
Invoker Identifier (IID)	Not present
Link Identifier (LID)	Link address of the specific DSRC-VU
Chaining	No
Element Identifier (EID)	As specified in the VST. No extension
Access Credentials	No
AttributeIdList	No extension, 1 attribute, AttributeID = 1 (RtmData)
Fragmentation	No
Layer2 settings	Command PDU, Polled ACn command

Table 14.11 shows an example of reading the RTM data.

**Table 14.11 — Presentation – Get Request frame example**

<i>Octet #</i>	<i>Attribute/Field</i>	<i>Bits in octet</i>	<i>Description</i>
1	FLAG	0111	Start flag
2	Private LID	xxxx	Link address of the specific DSRC-
3		xxxx	VU
4		xxxx	
5		xxxx	
6	MAC Control field	1010 s000	Command PDU
7	LLC Control field	n111	Polled ACn command, n bit
8	Fragmentation header	1xxx	No fragmentation

<i>Octet #</i>	<i>Attribute/Field</i>	<i>Bits in octet</i>	<i>Description</i>
9	Get.request	0110	Get request
	SEQUENCE {		
	OPTION indicator	0	Access Credentials not present
	OPTION indicator	0	IID not present
	OPTION indicator	1	AttributeIdList present
	OPTION indicator	0	Set to 0.
10	EID	xxxx	The EID of the RTM application
	INTEGER(0..127,...)	xxxx	instance,
11	AttributeIdList SEQUENCE OF {	0000	No extension, number of attributes =
12		0000	AttributeId=1, RtmData. No
		0001	extension
13	FCS	xxxx	Frame check sequence
14		xxxx	
15	Flag	0111	End Flag

DSC\_50 The *DSRC-VU*, when receiving the GET request, sends a GET response with the requested data conforming to the GET response defined in EN 13372, 6.2, 6.3, 6.4 and EN 12834, with settings as specified in Table 14.12.

**Table 14.12 — Presentation - GET response frame settings**

<i>Field</i>	<i>Settings</i>
Invoker Identifier (IID)	Not present
Link Identifier (LID)	As per EN 12834
Chaining	No
Element Identifier	As specified in the VST.
Access Credentials	No
Fragmentation	No
Layer2 settings	Response PDU, Response available and command accepted, ACn command

Table 14.13 shows an example of reading the RTM data. **Table 14.13 — Presentation - Response frame contents example**

<i>Octet #</i>	<i>Attribute/Field</i>	<i>Bits in octet</i>	<i>Description</i>
1	FLAG	0111 1110	Start flag
2	Private LID	xxxx xxxx	Link address of the specific DSRC-
3		xxxx xxxx	VU
4		xxxx xxxx	

<i>Octet #</i>	<i>Attribute/Field</i>	<i>Bits in octet</i>	<i>Description</i>
5		xxxx xxxx	
6	MAC Control field	1101 0000	Response PDU
7	LLC Control field	n111 0111	Response available, ACn command
8	LLC Status field	0000 0000	Response available and command
9	Fragmentation header	1xxx x001	No fragmentation
10	Get.response	0111	Get response
	SEQUENCE {		
	OPTION indicator	0	IID not present
	OPTION indicator	1	Attribute List present
	OPTION indicator	0	Return status not present
	OPTION indicator	0	Not used
11	EID INTEGER(0..127,...)	xxxx xxxx	Responding from the RTM application
12	AttributeList SEQUENCE OF {	0000 0001	No extension, number of attributes = 1
13	Attributes SEQUENCE { AttributeId	0000 0001	No extension, AttributeId=1 (RtmData)
14	AttributeValue CONTAINER {	0000 1010	No extension, Container Choice = 1010.
15		kkkk kkkk	RtmData
16		kkkk kkkk	
17		kkkk kkkk	
...		...	
n		kkkk kkkk	
n+1	FCS	xxxx xxxx	Frame check sequence
n+2		xxxx xxxx	
n+3	Flag	0111 1110	End Flag

DSC\_51 The REDCR then closes the connection by issuing a EVENT\_REPORT, RELEASE command conforming to EN 13372, 6.2, 6.3, 6.4 and EN 12834 ,7.3.8, with no specific RTM settings. Table 14.14 shows a bit encoding example of the RELEASE command.

**Table 14.14 — Termination. EVENT\_REPORT Release frame contents**

<i>Octet #</i>	<i>Attribute/Field</i>	<i>Bits in octet</i>	<i>Description</i>
1	FLAG	0111 1110	Start flag
2	Private LID	xxxx xxxx	Link address of the specific
3		xxxx xxxx	DSRC-VU
4		xxxx xxxx	



Octet #	Attribute/Field	Bits in octet	Description
5		xxxx xxxx	
6	MAC Control field	1000 s000	The frame contains a command
7	LLC Control field	0000 0011	UI command
8	Fragmentation header	1xxx x001	No fragmentation
9	EVENT_REPORT.request	0010	EVENT_REPORT (Release)
	SEQUENCE 1		
	OPTION indicator	0	Access Credentials not present
	OPTION indicator	0	Event parameter not present
	OPTION indicator	0	IID not present
	OPTION indicator	0	No response expected
10	EID            INTEGER	0000 0000	No extension, EID = 0 (System)
11	EventType    INTEGER	0000 0000	Event type 0 = Release
12	FCS	xxxx xxxx	Frame check sequence
13		xxxx xxxx	
14	Flag	0111 1110	End Flag

DSC\_52 The *DSRC-VU* is not expected to answer to the Release command. The communication is then closed.

#### 4.4.104 DSRC Test transaction description

DSC\_53 Full tests that include securing the data, need to be carried out as defined in Appendix 11 Common Security Mechanisms, by authorised persons with access to security procedures, using the normal GET command as defined above.

DSC\_54 Commissioning and periodic inspection tests that require decrypting and comprehension of the decrypted data content shall be undertaken as specified in Appendix 11 Common Security Mechanisms and Appendix 9, Type Approval List of Minimum required tests.

However, the basic DSRC communication can be tested by the command ECHO. Such tests may be required on commissioning, at periodic inspection, or otherwise to the requirement of the competent control authority or Regulation (EU) N°. 165/2014 (See 6 below)

DSC\_55 In order to effect this basic communication test, the ECHO command is issued by the REDCR during a session, i.e., after an initialisation phase has been completed successfully. The sequence of interactions is thus similar to that of an interrogation:

↳ Step 1 The REDCR sends a 'beacon service table' (BST) that includes the application identifiers (AIDs) in

the service list that it supports. In the RTM applications this will simply be the service with the AID value = 2.

- ↪ The *DSRC-VU* evaluates the received BST, and where it identifies that the BST is requesting Freight&Fleet (AID = 2), the *DSRC-VU* shall respond. If *the REDCR* does not offer AID=2, the *DSRC-VU* shall shut down its transaction with *the REDCR*.
- ↪ Step 2 *The DSRC-VU* sends a request for a private window allocation.
- ↪ Step 3 *The REDCR* sends a private window allocation.
- ↪ Step 4 *The DSRC-VU* uses the allocated private window to send its vehicle service table (VST). This VST includes a list of all the different application instantiations that this *DSRC-VU* supports in the framework of AID=2. The different instantiations shall be identified by means of uniquely EIDs, each associated with a parameter value indicating the instance of the application that is supported.
- ↪ Step 5 Next *the REDCR* analyses the offered VST, and either terminates the connection (RELEASE) since it is not interested in anything the VST has to offer (i.e., it is receiving a VST from a *DSRC-VU* that is not an RTM VU, or, if it receives an appropriate VST it starts an app instantiation.
- ↪ Step 6 *The REDCR* shall issue a command (ECHO) to the specific *DSRC-VU*, and allocates a private window.
- ↪ Step 7 *The DSRC-VU* uses the newly allocated private window to send an ECHO response

frame.

The following tables give a practical example of an ECHO exchange session.

DSC\_56 Initialisation is performed according to 5.4.7 (DSC\_44 – DSC\_48) and Tables 14.4 – 14.9

DSC\_57 The REDCR then issues an ACTION, ECHO command conforming to ISO 14906, containing 100 octets of data and with no specific settings for RTM. Table 14.15 shows the contents of the frame sent by the REDCR.

**Table 14.15 — ACTION, ECHO request frame example**

Octet #	Attribute/Field	Bits in octet	Description
1	FLAG	0111 1110	Start flag
2	Private LID	xxxx xxxx	Link address of the specific
3		xxxx xxxx	DSRC-VU
4		xxxx xxxx	
5		xxxx xxxx	
6	MAC Control field	1010 s000	Command PDU
7	LLC Control field	n111 0111	Polled ACn command, n bit
8	Fragmentation header	1xxx x001	No fragmentation
9	ACTION.request	0000	Action request (ECHO)
	SEQUENCE #		
	OPTION indicator	0	Access Credentials not present
	OPTION indicator	1	Action parameter present
	OPTION indicator	0	IID not present
	Mode	BOOLEAN	1 Response expected

Octet #	Attribute/Field		Bits in octet	Description
10	EID	INTEGER	0000 0000	No extension, EID = 0 (System)
11	ActionType	INTEGER	0000 1111	No extension, Action type ECHO
12	ActionParameter	CONTAINER {	0000 0010	No extension, Container Choice =
13			0110 0100	No extension. String length = 100
14			xxxx xxxx	Data to be echoed
...			...	
11			xxxx xxxx	
11	FCS		xxxx xxxx	Frame check sequence
11			xxxx xxxx	
11	Flag		0111 1110	End Flag

DSC\_58 The *DSRC-VU*, when receiving the ECHO request, sends an ECHO response of 100 octets of data by reflecting the received command, according to ISO 14906, with no specific settings for RTM. Table 14.16 shows a bit level encoding example. **Table 14.16 — ACTION, ECHO response frame example**

Octet #	Attribute/Field		Bits in octet	Description
1	FLAG		0111 1110	Start flag
2	Private LID		xxxx xxxx	Link address of the specific VU
3			xxxx xxxx	
4			xxxx xxxx	
5			xxxx xxxx	
6	MAC Control field		1101 0000	Response PDU
7	LLC Control field		n111 0111	ACn command n bit
8	LLC status field		0000 0000	Response available
9	Fragmentation header		1xxx x001	No fragmentation
10	ACTION.response		0001	ACTION response (ECHO)
	SEQUENCE 1			
	OPTION indicator		0	IID not present
	OPTION indicator		1	Response parameter present
	OPTION indicator		0	Return status not present
	Fill BIT STRING (SIZE (1))		0	Not used
11	EID	INTEGER (0..127,...)	0000 0000	No extension, EID = 0 (System)
12	ResponseParameter	CONTAINER {	0000 0010	No extension, Container Choice =
13			0110 0100	No extension. String length = 100
14			xxxx xxxx	Echoed data
...			....	
113	}}		xxxx xxxx	
114	FCS		xxxx xxxx	Frame check sequence
115			xxxx xxxx	
116	Flag		0111 1110	End Flag

## 4.5 Support for Directive 2015/71/EC

### 4.5.1 Overview

DSC\_59 To support the Directive 2015/719/EC on the maximal weights and dimensions for heavy goods vehicles, the transaction protocol to download OWS data across the 5.8 GHz DSRC interface link will be the same as that used for the RTM data (see 4.4.1), the only difference being that the Object Identifier that relates to the TARV standard will be addressing the ISO 15638 standard (TARV) Part 20 related to WOB/OWS.

### 4.5.2 Commands

DSC\_60 The commands used for an OWS transaction will be the same as those used for an RTM transaction.

### 4.5.3 Interrogation command sequence

DSC\_61 The interrogation command sequence for OWS data will be the same as for RTM data.

### 4.5.4 Data structures

DSC\_62 The payload (OWS data) consists of the concatenation of

1. EncryptedOwsPayload data, which is the encryption of the OwsPayload defined in ASN.1 in section 4.5.5. The method of encryption shall be the same adopted for the RtmData, which is specified in Appendix 11
2. DSRCSecurityData, calculated with the same algorithms adopted for the RtmData, which is specified in Appendix 11.

### 4.5.5 ASN.1 module for the OWS DSRC transaction

DSC\_63. The ASN.1 module definition for the DSRC data within the RTM application is defined as follows:

```
TarvOws {iso(1) standard(0) 15638 part20(20)
version1(1)} DEFINITIONS AUTOMATIC TAGS
 ::= BEGIN
IMPORTS
-- Imports data attributes and elements from EFC which are used for OWS
LPN
FROM EfcDsrcApplication {iso(1) standard(0) 14906 application(0) version5(5)}

-- Imports function parameters from the EFC Application Interface Definition
SetMMIRq
FROM EfcDsrcApplication {iso(1) standard(0) 14906 application(0) version5(5)}

-- Imports the L7 DSRCData module data from the EFC Application Interface Definition
Action-Request, Action-Response, ActionType, ApplicationList, AttributeIdList, AttributeList, Attributes,
BeaconID, BST, Dsrc-EID, DSRCApplicationEntityID, Event-Report-Request, Event-Report- Response,
EventType, Get-Request, Get-Response, Initialisation-Request, Initialisation-Response,
ObeConfiguration, Profile, ReturnStatus, Time, T-APDUs, VST
```

FROM EfcDsrcGeneric {iso(1) standard(0) 14906 generic(1) version5(5)};

-- Definitions of the OWS functions:

OWS-InitialiseComm-Request ::= BST

OWS-InitialiseComm-Response ::= VST

OWS-DataRetrieval-Request ::= Get-Request (WITH COMPONENTS {fill (SIZE(1)), eid, accessCredentials ABSENT, iid ABSENT, attrIdList})

OWS-DataRetrieval-Response ::= Get-Response {OwsContainer} (WITH COMPONENTS {..., eid, iid ABSENT})

OWS-TerminateComm ::= Event-Report-Request {OwsContainer} (WITH COMPONENTS {mode (FALSE), eid (0), eventType (0)})

OWS-TestComm-Request ::= Action-Request {OwsContainer} (WITH COMPONENTS {..., eid (0), actionType (15), accessCredentials ABSENT, iid ABSENT})

OWS-TestComm-Response ::= Action-Response {OwsContainer} (WITH COMPONENTS {..., fill (SIZE(1)), eid (0), iid ABSENT})

-- Definitions of the OWS attributes:

OwsData ::= SEQUENCE {

    encryptedOwsPayload OCTET STRING (SIZE(51)) (CONSTRAINED BY { -- calculated encrypting OwsPayload as per Appendix 11 --}),

    DSRCSecurityData OCTET STRING

}

OwsPayload ::= SEQUENCE {

    tp15638VehicleRegistrationPlate LPN -- Vehicle Registration Plate as per EN 15509.

    recordedWeight INTEGER (0..65535), -- 0= Total measured weight of the heavy goods vehicle  
    -- with 10 Kg resolution.

    axlesConfiguration OCTET STRING SIZE (3), -- 0= 20 bits allowed for the number  
    -- of axles for 10 axles.

    axlesRecordedWeight OCTET STRING SIZE (20), -- 0= Recorded Weight for each axle  
    -- with 10 Kg resolution.

    tp15638Timestamp INTEGER(0..4294967295) -- Timestamp of current record

}

Ows-ContextMark ::= SEQUENCE {

    standardIdentifier StandardIdentifier, -- identifier of the TAR V part and its version

}

StandardIdentifier ::= OBJECT IDENTIFIER

OwsContainer ::= CHOICE {

    integer [0] INTEGER,

    bitstring [1] BIT STRING,

```

    octetstring      [2]  OCTET STRING (SIZE (0..127, ...)),
    universalString  [3]  UniversalString,
    beaconId        [4]  BeaconID,
    t-apdu          [5]  T-APDUs,
    dsrcApplicationEntityId [6] DSRCApplicationEntityID,
    dsrc-Ase-Id     [7]  Dsrc-EID,
    attrIdList      [8]  AttributeIdList,
    attrList        [9]  AttributeList{RtmContainer},
    reserved10      [10] NULL,
    OwsContextmark  [11] Ows-ContextMark,
    OwsData         [12] OwsData,
    reserved13      [13] NULL,
    reserved14      [14] NULL,
    time           [15] Time,
-- values from 16 to 255 reserved for ISO/CEN usage
    }}

```

END

#### 4.5.6 Elements of OwsData, actions performed and definitions

The elements of OwsData are defined to support Directive 2015/719/EC on the maximal weights and dimensions for heavy goods vehicles. Their meaning is:

- recordedWeight represents the total measured weight of the heavy goods vehicle with a resolution of 10 Kg as defined in EN ISO 14906. For example, a value of 2500, represent a weight of 25 tons.
- axlesConfiguration represents the configuration of the heavy goods vehicle as number of axles. The configuration is defined with the bit mask of 20 bits (extended from EN ISO 14906).

A bit mask of 2 bits represents the configuration of an axle with the following format:

- Value 00B means that value is "non available" because the vehicle does not have equipment to collect the weight on the axle.
- Value 01B means that the axle is not present.
- Value 10B means that the axle is present and the weight has been calculated and collected and it is provided in the axlesRecordedWeight field.
- Value 11B is reserved for future uses.

The last 4 bits are reserved for future uses.

*Number of Axles*

Number of axles on tractor unit			Number of axles on trailer							
00/01/10/ /11	00/01/10/ 11	00/01/10/ 11	00/0 1/10/ 11	00/0 1/10/ 11	00/0 1/10/ 11	00/0 1/10/ 11	00/0 1/10/ 11	00/0 1/10/ 11	00/0 1/10/ 11	RFU (4 bits)

- `axlesRecordedWeight` represent the specific weight recorded for each axle with a resolution of 10 Kg. Two octets are used for each axle. For example, a value of 150, represent a weight of 1500 Kgs.

The other data types are defined in 4.4.101.

#### 4.5.7 Data transfer mechanisms

DSC\_64 The Data transfer mechanism for OWS data between the Interrogator and the DSRC facility in the vehicle shall be the same as for RTM data (see 4.4.102).

DSC\_65 The Data transfer between the platform collecting the maximal weights data and the DSRC facility in the vehicle shall be based on the physical connection and interfaces and protocol defined in section 4.6.

#### 4.6 Data transfer between the DSRC-VU and VU

##### 4.6.1 Physical Connection and interfaces

DSC\_66 The connection between the *VU* and the *DSRC-VU* can be either by physical cable or short range wireless communication based on Bluetooth v4.0 BLE.

DSC\_67 Regardless of the choice of the physical connection and interface, the following requirements shall be satisfied:

DSC\_68 a) In order that different suppliers may be contracted to supply the *VU* and the *DSRC-VU*, and indeed different batches of *DSRC-VU*, the connection between the *VU* and the *DSRC-VU* shall be an open standard connection. The *VU* shall connect with the *DSRC-VU* either

- i) using fixed cable of at least 2 meters, using a Straight DIN 41612 H11 Connector – 11 pin approved male connector from the *DSRC-VU* to match a similar DIN/ISO approved female connector from the *VU* device,
- ii) using Bluetooth Low Energy (BLE)
- iii) using a standard ISO 11898 or SAE J1939 connection

DSC\_69 b) the definition of the interfaces and connection between the *VU* and *DSRC-VU* must support the application protocol commands defined in 5.6.2. and

DSC\_70 c) the *VU* and *DSRC-VU* must support the operation of the data transfer via the connection in regard to performance and power supply.

## 4.6.2 Application Protocol

DSC\_71 The application protocol between the VU Remote Communication facility and DSRC-VU is responsible for periodically transferring the remote communication data from the VU to the DSRC.

DSC\_72 The following main commands are identified:

1. Initialisation of the communication link - Request
2. Initialisation of the communication link – Response
3. Send Data with Identifier of the RTM application and Payload defined by RTM Data
4. Acknowledgment of the data
5. Termination of the communication link - Request
6. Termination of the communication link - Response

DSC\_73 In ASN1.0, the previous commands may be defined as:

```
Remote Communication DT Protocol DEFINITIONS ::= BEGIN

    RCDT-CommunicationLink Initialization - Request ::= SEQUENCE { LinkIdentifier INTEGER
    }

    RCDT-CommunicationLink Initialization - Response ::= SEQUENCE { LinkIdentifier INTEGER,
        answer      BOOLEAN
    }

    RCDT- Send Data ::= SEQUENCE {
        LinkIdentifier INTEGER,
        DataTransactionId INTEGER, RCDTData
        SignedTachographPayload
    }

    RCDT Data Acknowledgment ::= SEQUENCE {
        LinkIdentifier INTEGER, DataTransactionId
        INTEGER,
        answer      BOOLEAN
    }

    RCDT-CommunicationLink Termination - Request ::= SEQUENCE { LinkIdentifier INTEGER
    }

    RCDT-CommunicationLink Termination - Response ::= SEQUENCE { LinkIdentifier
        INTEGER,
        answer      BOOLEAN
    }

End
```

DSC\_74 The description of the commands and parameters is following:

- RCDT-Communication Link Initialization - Request is used to initialize the communication link. The command is sent by the VU to the DSRC-VU. The LinkIdentifier is set by the VU and communicated to the DSRC-VU to track a specific communication link.



(Note: this is to support future links and other application/modules like Weighing on board).

- RCDT-Communication Link Initialization - Response is used by the DSRC-VU to provide the response of the request to initialize the communication link. The command is sent by the DSRC-VU to the VU. The command provides the result of the initialisation as answer = 1 (Success) or =0 (Failure).

DSC\_75 The initialization of the communication link shall be done only after installation, calibration, and start of the engine/VU

is switched on.

- RCDT-Send Data is used by the VU to send the signed RCDTData (i.e., *the remote communication Data*) to the DSRC-VU. The data will be sent every 60 seconds. The DataTransactionId parameter identifies the specific transmission of data. The LinkIdentifier is also used to ensure that the appropriate link is correct.
- RCDT-Data Acknowledgment is sent by the DSRC-VU to provide the feedback to the VU on the reception of the data from a RCDT-Send Data command identified by the DataTransactionId parameter. The Answer parameter is 1 (Success) or =0 (Failure). If a VU receives more than three answers equal to 0 or if the VU does not receive a RCDT Data Acknowledgment for a specific previously sent RCDT-Send Data with a specific DataTransactionId, the VU will generate and record an event.
- RCDT-Communication Link Termination request is sent by the VU to DSRC-VU to terminate a link for a specific LinkIdentifier.

DSC\_76 At the restart of the DSRC-VU or a VU, all the existing Communication Links should be removed as there could be “dangling” Links due to the sudden shutdown of a VU.

- RCDT-Communication Link Termination - Response is sent by the DSRC-VU to the VU to confirm the request of termination of the link by the VU for the specific LinkIdentifier.

## 4.7 Error handling

### 4.7.1 Recording and communication of the Data in the DSRC-VU

DSC\_77 The Data shall be provided, already secured, by the VUSM function to the DSRC-VU. The VUSM shall verify that data recorded in the DSRC-VU has been recorded correctly. The recording and reporting of any errors in the transfer of data from the VU to the memory of the DSRC-VU shall be recorded with type EventFaultType and enum value set to ‘62’H Remote Communication Facility communication fault together with the timestamp.

DSC\_78 The VU shall maintain a file identified by a unique name that is easily identifiable by inspectors for the purpose of recording “VU internal communication failures”.

DSC\_79 If the VUPM attempts to obtain VU data from the security module (to pass to the VU-DSRC), but fails to do so, it shall record that failure with type EventFaultType and enum value set to ‘62’H Remote Communication Facility communication fault together with the timestamp. The failure of the communication is detected when a RCDT Data Acknowledgment message is not received for the related (i.e., with the same DataTransactionId in the Send Data and Acknowledgment messages) RCDT Send Data for more than three consecutive times.

### 4.7.2 Wireless Communication errors

DSC\_80 Communication error handling shall be consistent with the related DSRC standards, namely EN 300 674-1, EN 12253, EN 12795, EN 12834 and the appropriate parameters of EN 13372.

## 4.7.2.1 Encryption and signature errors

DSC\_81 Encryption and signature errors shall be handled as defined in Appendix 11 Common Security Mechanisms and are not present in any error messages associated with the DSRC transfer of data.

## 4.7.2.2 Recording of errors

The DSRC medium is a dynamic wireless communication in an environment of uncertain atmospheric and interference conditions, particularly in the 'portable REDCR and 'moving vehicle' combinations involved in this application. It is therefore necessary to ascertain the difference between a 'read failure' and an 'error' condition. In a transaction across a wireless interface, read failure is common and the consequence is usually to retry, i.e. rebroadcast the BST and reattempt the sequence, which will in most circumstances lead to a successful communication connection and transfer of data, unless the target vehicle moves out of range during the time required to retransmit. (A 'successful' instance of a 'read' may have involved several attempts and retries).

Read failure may be because the antennas were not paired properly (failure of 'aiming'); because one of the antennas is shielded – this may be deliberate, but also can be caused by the physical presence of another vehicle; radio interference, especially from circa 5.8 GHz WIFI or other public access wireless communications, or may be caused by radar interference, or difficult atmospheric conditions (e.g. during a thunderstorm); or simply by moving out of the range of the DSRC communication. Individual instances of read failures, by their nature, cannot be recorded, simply because the communication simply did not occur.

However, if the agent of the competent control authority targets a vehicle and attempts to interrogate its *DSRC-VU*, but no successful transfer of data ensues, this failure could have occurred because of deliberate tampering, and therefore the agent of the competent control authority needs a means to log the failure, and alert colleagues downstream that there may be a violation. The colleagues can then stop the vehicle and carry out a physical inspection. However, as no successful communication has taken place, the *DSRC-VU* cannot provide data concerning the failure. Such reporting shall therefore be a function of REDCR equipment design.

'Failure to read' is technically different to an 'error'. In this context an 'error' is the acquisition of a wrong value.

Data transferred to the *DSRC-VU* is supplied already secured, therefore must be verified by the supplier of the data (see 5.4).

Data subsequently transferred across the air interface is checked by cyclic redundancy checks at the communications level. If the CRC validates, then the data is correct. If the CRC does not validate, the data is retransmitted. The probability that data could successfully pass through a CRC incorrectly is statistically so highly improbable that it may be discounted.

If the CRC does not validate and there is no time to retransmit and receive the correct data, then the result will not be an error, but an instantiation of a specific type of read failure.

The only meaningful 'failure' data that can be recorded is that of the number of successful initiations of transactions that occur, that do not result in a successful transfer of data to the REDCR.

DSC\_82 The *REDCR* shall therefore record, time-stamped, the number of occasions where the 'initialisation' phase of a DSRC interrogation is successful, but the transaction terminated before *the Data* was successfully retrieved by the REDCR. This data shall be available to agent of the competent control authority and shall be stored in the memory of the REDCR equipment. The means by which this is achieved shall be a matter of product design or the specification of a competent control authority.

The only meaningful ‘error’ data that can be recorded is the number of occasions where the REDCR fails to decrypt *the Data* received. However, it should be noted that this will only relate to the efficiency of the REDCR software. Data may be technically decrypted, but make no semantic sense.

DSC\_83 The REDCR shall therefore record, time-stamped, the number of occasions where it has attempted but failed to decipher data received across the DSRC interface.

## 5 Commissioning and periodic inspection tests for the remote communication function

### 5.1 General

DSC\_84 Two type of tests are foreseen for the remote communication function:

- 1) An ECHO test to validate the *DSRC-REDCR >>:-<DSRC-VU wireless* communication channel.
- 2) A End-to-end security test to ensure that a workshop card is able to access the encrypted and signed data content created by the VU and transmitted over the wireless communication channel.

### 5.2 ECHO

This clause contains provisions specifically made to test only that the *DSRC-REDCR >>:-<DSRC-VU* is functionally active.

The objective of the ECHO command is to enable workshops or type approval test facilities to test that the DSRC link is working without needing access to security credentials. The tester’s equipment therefore only needs to be able to initialise a DSRC communication (sending a BST with AID=2) and then send the ECHO command, and, assuming the DSRC is working, will receive the ECHO response. See 5.4.8 for details. Assuming it receives this response correctly, the DSRC link (*DSRC-REDCR >>:-<DSRC-VU*) may be validated as functioning correctly.

### 5.3 Tests to validate the secure data content

DSC\_85 This test is execute to validate the end-to-end security flow of data. A DSRC test reader is needed for such test. The DSRC test reader performs the same functionality and it is implemented with the same specifications of the reader used by the law enforcers, with the difference that a workshop card shall be used to authenticate the user of the DSRC test reader rather than a control card. The test can be executed after the initial activation of a Smart Tachograph or at the end of the calibration procedure. After the activation, the vehicle unit shall generate and communicate to the DSRC-VU the secured early detection data.

DSC\_86 The workshop personnel must position the DSRC test reader at a distance between 2 and 10 metres in front of the vehicle.

DSC\_87 Then the workshop personnel will insert a workshop card in the DSRC test reader to request the interrogation of the early detection data to the vehicle unit. After a successful interrogation, the workshop personnel will access the received data to ensure that it has been successfully validated for integrity and decrypted.

## Appendix 15

# MIGRATION: Managing the co-existence of equipment generations

### TABLE OF CONTENT

<b>1. DEFINITIONS</b> _____	<b>613</b>
<b>2. GENERAL PROVISIONS</b> _____	<b>613</b>
2.1. Overview of the transition.....	613
2.2. Interoperability between VU and cards.....	613
2.3. Interoperability between VU and MS.....	614
2.4. Interoperability between vehicle units, tachograph cards and equipment for data download.....	616
2.4.1 Direct card download by IDE.....	614
2.4.2 Card download through a vehicle unit.....	614
2.4.3 Vehicle unit download.....	614
2.5. Interoperability between VU and calibration equipment.....	<b>615</b>
<b>3. MAIN STEPS DURING THE PERIOD BEFORE THE INTRODUCTION DATE</b> _____	<b>615</b>
<b>4. PROVISIONS FOR THE PERIOD AFTER THE INTRODUCTION DATE</b> _____	<b>615</b>

## 1. Definitions

For the purposes of this Appendix, the following definitions are used.

**smart tachograph system:** as defined by this Annex (chapter 1: definition bbb);

**first generation tachograph system:** as defined by this Regulation (article 2: definition 1);

**second generation tachograph system:** as defined by this Regulation (article 2: definition 7);

**introduction date:** as defined by this Annex (chapter 1: definition ccc);

**Intelligent Dedicated Equipment (IDE):** equipment used to perform data downloading, as defined in Appendix 7 of this Annex.

## 2. General Provisions

### 2.1. Overview of the transition

The preamble of this Annex provides an overview of the transition between the first and the second generation tachograph systems.

In addition to the provisions of this preamble:

- first generation motion sensors will not be interoperable with second generation vehicle units.
- second generation motion sensors will start to be installed in vehicles at the same time as second generation vehicle units.
- data download and calibration equipment will need to evolve, in order to support use of both generation of recording equipment and tachograph cards.

### 2.2. Interoperability between VU and cards

It is understood that first generation tachograph cards are interoperable with first generation vehicle units (in compliance with Annex 1B of this Regulation), while second generation tachograph cards are interoperable with second generation vehicle units (in compliance with Annex 1C of this Regulation). In addition, the requirements below shall apply.

- MIG\_001 Except as provided for in requirement MIG\_004 and MIG\_005, first generation tachograph cards may continue to be used in second generation vehicle units until their end of validity date. Their holders may however ask for their replacement by second generation tachograph cards as soon as they are available.
- MIG\_002 Second generation vehicle units shall be able to use any valid first generation driver, control and company card inserted.
- MIG\_003 This capability may be suppressed once and forever in such vehicle units by workshops, so that first generation tachograph cards cannot be accepted anymore. This may only be done after the European Commission has launched a procedure aiming to request workshops to do so, for example during each periodic inspection of tachograph.
- MIG\_004 Second generation vehicle units shall only be able to use second generation workshop cards.
- MIG\_005 For determining the mode of operation, second generation vehicle units shall only consider the types of the valid cards inserted, regardless of their generations.
- MIG\_006 Any valid second generation tachograph card shall be able to be used in first generation vehicle units exactly the same manner as a first generation tachograph card of the same type.

### 2.3. Interoperability between VU and MS

It is understood that first generation motion sensors are interoperable with first generation vehicle units, while second generation motion sensors are interoperable with second generation vehicle units. In addition, the requirements below shall apply.

MIG\_007 Second generation vehicle units will not be able to be paired and used with first generation motion sensors.

MIG\_008 Second generation motion sensors may be paired and used with second generation vehicle units only, or with both generations of vehicle units.

### 2.4. Interoperability between vehicle units, tachograph cards and equipment for data download

MIG\_009 Equipment for data download may be used with one generation only of vehicle units and tachograph cards, or with both.

#### 2.4.1 Direct card download by IDE

MIG\_010 Data shall be downloaded by IDE from tachograph cards of one generation inserted in their card readers, using the security mechanisms and the data download protocol of this generation, and downloaded data shall have the format defined for this generation.

MIG\_011 To allow drivers' control by non EU control authorities, it shall also be possible to download second generation driver (and workshop) cards exactly the same manner as 1<sup>st</sup> generation drivers (and workshop) cards. Such download shall include:

- non signed EFs IC and ICC,
- non signed EFs (1<sup>st</sup> generation) Card\_Certificate and CA\_Certificate,
- the other application data EFs (within TACHO DF) requested by the first generation card download protocol. This information shall be secured with a digital signature, according to the first generation security mechanisms.

Such download shall not include application data EFs only present in second generation driver (and workshop) cards (application data EFs within TACHO\_G2 DF).

#### 2.4.2 Card download through a vehicle unit

MIG\_012 Data shall be downloaded from a second generation card inserted in a first generation vehicle unit using the first generation data download protocol. The card shall answer to the vehicle unit commands exactly the same manner as a first generation card and downloaded data shall have the same format as data downloaded from a first generation card.

MIG\_013 Data shall be downloaded from a first generation card inserted in a second generation vehicle unit using the data download protocol defined in Appendix 7 of this Annex. The vehicle unit shall send commands to the card exactly the same manner as a first generation vehicle unit, and downloaded data shall respect the format defined for first generation cards.

#### 2.4.3 Vehicle unit download

MIG\_014 Data shall be downloaded from second generation vehicle units using the second generation security mechanisms, and the data download protocol specified in Appendix 7 of this Annex.

MIG\_015 To allow drivers' control by non EU control authorities and vehicle unit data download by non EU workshops, it may optionally also be possible to download data from second generation vehicle units using the first generation security mechanisms, and the first generation data download protocol. Downloaded data shall have the same format as data downloaded from a first generation vehicle unit. This capability may be selected through commands in the menu.

2.5. Interoperability between VU and calibration equipment

MIG\_016 Calibration equipment shall be able to perform calibration of each generation of tachograph, using the calibration protocol of this generation. Calibration equipment may be used with one generation only of tachograph, or with both.

3. Main steps during the period before the introduction date

MIG\_017 Test keys and certificates shall be available to manufacturers at the latest **30 months** before the introduction date.

MIG\_018 Interoperability tests shall be ready to start if requested by manufacturers at the latest **15 months** before the introduction date.

MIG\_019 Official keys and certificates shall be available to manufacturers at the latest **12 months** before the introduction date.

MIG\_020 Member states shall be able to issue second generation workshop cards at the latest **3 months** before the introduction date.

MIG\_021 Member States shall be able to issue all types of second generation tachograph cards at the latest **1 month before the introduction date**.

4. Provisions for the period after the introduction date

MIG\_022 After the introduction date, Member States shall only issue second generation tachograph cards.

MIG\_023 Vehicle units / motion sensors manufacturers shall be allowed to produce first generation vehicle units / motion sensors as long as they are used in the field, so that malfunctioning components can be replaced.

MIG\_024 Vehicle units / motion sensors manufacturers shall be allowed to request and obtain type approval maintenance of first generation vehicle units / motion sensors types already type approved.

## APPENDIX 16 ADAPTOR FOR M 1 AND N1 CATEGORY VEHICLES

### Table of Contents

<b>1. Abbreviations and reference documents</b>	<b>617</b>
1.1. .... Abbreviations.....	617
1.2. .... Reference standards .....	617
<b>2. General characteristics and functions of the adaptor</b>	<b>617</b>
2.1. .... Adaptor general description .....	617
2.2. .... Functions.....	617
2.3. .... Security .....	617
<b>3. Requirements for the recording equipment when an adaptor is installed</b>	<b>617</b>
<b>4. Construction and functional requirements for the adaptor</b>	<b>618</b>
4.1. .... Interfacing and adapting incoming speed pulses .....	618
4.2. .... Inducing the incoming pulses to the embedded motion sensor.....	618
4.3. .... Embedded motion sensor .....	618
4.4. .... Security requirements .....	618
4.5. .... Performance characteristics .....	619
4.6. .... Materials .....	619
4.7. .... Markings .....	619
<b>5. Installation of the recording equipment when an adaptor is used</b>	<b>619</b>
5.1. .... Installation .....	619
5.2. .... Sealing .....	620
<b>6. Checks, inspections and repairs</b>	<b>620</b>
6.1. .... Periodic inspections .....	620
<b>7. Type approval of recording equipment when an adaptor is used</b>	<b>620</b>
7.1. .... General points .....	620
7.2. .... Functional certificate .....	620



## 1. Abbreviations and reference documents

### 1.1. Abbreviations

TBD To Be Defined

VU Vehicle Unit

### 1.2. Reference standards

ISO16844-3 Road vehicles – Tachograph systems – Part 3: Motion sensor interface

## 2. General characteristics and functions of the adaptor

### 2.1. Adaptor general description

ADA\_001 The adaptor shall provide a connected VU with secured motion data permanently representative of vehicle speed and distance travelled.

The adaptor is only intended for those vehicles that are required to be equipped with recording equipment in compliance with this Regulation.

It shall be installed and used only in those types of vehicle defined in definition yy) ‘adaptor’ of Annex IC where it is not mechanically possible to install any other type of existing motion sensor which is otherwise compliant with the provisions of this Annex and its Appendixes 1 to 16.

The adaptor shall not be mechanically interfaced to a moving part of the vehicle, but connected to the speed/distance impulses which are generated by integrated sensors or alternative interfaces.

ADA\_002 A type approved motion sensor (according to the provisions of this Annex IC, section 8, Type approval of recording equipment and tachograph cards) shall be fitted into the adaptor housing, which shall also include a pulse converter device inducing the incoming pulses to the embedded motion sensor. The embedded motion sensor itself shall be connected to the VU, so that the interface between the VU and the adaptor shall be compliant with the requirements set out in ISO16844-3.

### 2.2. Functions

ADA\_003 The adaptor shall include the following functions:

- interfacing and adapting the incoming speed pulses,
- inducing the incoming pulses to the embedded motion sensor,
- all functions of the embedded motion sensor, providing secured motion data to the VU.

### 2.3. Security

ADA\_004 The adaptor shall not be security certified according to the motion sensor generic security target defined in Appendix 10 of this Annex. Security related requirements specified in section 4.4 of this Appendix shall apply instead.

## 3. Requirements for the recording equipment when an adaptor is installed

The requirements in the following Chapters indicate how the requirements of this Annex shall be understood when an adaptor is used. The related requirement numbers of Annex IC are provided between brackets.

ADA\_005 The recording equipment of any vehicle fitted with an adaptor must comply with all the provisions of this Annex, except otherwise specified in this Appendix.

ADA\_006 When an adaptor is installed, the recording equipment includes cables, the adaptor (including a motion sensor), and a VU [01].

ADA\_007 The detection of events and/or faults function of the recording equipment is modified as follows:

- the “power supply interruption” event shall be triggered by the VU, while not in calibration mode, in case of any interruption exceeding 200 milliseconds of the power supply of the embedded motion sensor [79]
- the “motion data error” event shall be triggered by the VU in case of interruption of the normal data flow between the embedded motion sensor and the VU and/or in case of data integrity or data authentication error during data exchange between the embedded motion sensor and the VU [83]
- the “security breach attempt” event shall be triggered by the VU for any other event affecting the security of the embedded motion sensor, while not in calibration mode [85]
- the “recording equipment” fault shall be triggered by the VU, while not in calibration mode, for any fault of the embedded motion sensor [88]

ADA\_008 The adaptor faults detectable by the recording equipment shall be those related with the embedded motion sensor [88].

ADA\_009 The VU calibration function shall allow to automatically pair the embedded motion sensor with the VU [202, 204].

#### 4. Construction and functional requirements for the adaptor

##### 4.1. Interfacing and adapting incoming speed pulses

ADA\_011 The adaptor input interface shall accept frequency pulses representative of the vehicle speed and distance travelled. Electrical characteristics of the incoming pulses are: *TBD by the manufacturer*. Adjustments accessible to only the adaptor manufacturer, and to the approved workshop performing the adaptor installation shall allow the correct interfacing of the adaptor input to the vehicle, if applicable.

ADA\_012 The adaptor input interface shall be able, if applicable, to multiply or divide the frequency pulses of the incoming speed pulses by a fixed factor, to adapt the signal to the k factor range defined by this Annex (4000 to 25000 pulses/km). This fixed factor may only be programmed by the adaptor manufacturer, and the approved workshop performing the adaptor installation.

##### 4.2. Inducing the incoming pulses to the embedded motion sensor

ADA\_013 The incoming pulses, possibly adapted as specified above, shall be induced to the embedded motion sensor, so that each incoming pulse shall be detected by the motion sensor.

##### 4.3. Embedded motion sensor

ADA\_014 The embedded motion sensor shall be stimulated by the induced pulses, thus allowing it to generate motion data accurately representing the vehicle movement, as if it was mechanically interfaced to a moving part of the vehicle.

ADA\_015 The identification data of the embedded motion sensor shall be used by the VU to identify the adaptor [95].

ADA\_016 The installation data stored in the embedded motion sensor shall be considered to represent the adaptor installation data [122].

##### 4.4. Security requirements

ADA\_017 The adaptor housing shall be designed so that it cannot be opened. It shall be sealed, so that physical tampering attempts can be easily detected (e.g. through visual inspection, see ADA\_035). Seals shall follow the same requirements of motion sensor seals [398 to 406]

ADA\_018 It shall not be possible to remove the embedded motion sensor from the adaptor without breaking the seal(s) of the adaptor housing, or breaking the seal between the sensor and the adaptor housing (see ADA\_034).

ADA\_019 The adaptor shall ensure that motion data may only be processed and derived from the adaptor input.

#### 4.5. Performance characteristics

ADA\_020 The adaptor shall be fully operational in the temperature range defined by the manufacturer.

ADA\_021 The adaptor shall be fully operational in the humidity range 10% to 90% [214].

ADA\_022 The adaptor shall be protected against over-voltage, inversion of its power supply polarity, and short circuits [216].

ADA\_023 The adaptor shall either:

react to a magnetic field disturbing vehicle motion detection. In such circumstances, the vehicle unit will record and store a sensor fault [88] or,

have a sensing element that is protected from, or immune to, magnetic fields [217].

ADA\_024 The adaptor shall conform to international regulation UN ECE R10, related to electromagnetic compatibility, and shall be protected against electrostatic discharges and transients [218].

#### 4.6. Materials

ADA\_025 The adaptor shall meet the protection grade (*TBD by the manufacturer, depending on the installation position*) [220, 221].

ADA\_026 The colour of the adaptor housing shall be yellow.

#### 4.7. Markings

ADA\_027 A descriptive plaque shall be affixed to the adaptor and shall show the following details:

- name and address of the manufacturer of the adaptor,
- manufacturer's part number and year of manufacture of the adaptor,
- approval mark of the adaptor type or of the recording equipment type including the adaptor,
- the date on which the adaptor has been installed,
- the vehicle identification number of the vehicle on which it has been installed.

ADA\_028 The descriptive plaque shall also show the following details (if not directly readable from the outside on the embedded motion sensor):

- name of the manufacturer of the embedded motion sensor,
- manufacturer's part number and year of manufacture of the embedded motion sensor,
- approval mark for the embedded motion sensor.

### 5. Installation of the recording equipment when an adaptor is used

#### 5.1. Installation

ADA\_029 Adaptors to be installed in vehicles shall only be installed by vehicle manufacturers, or by approved workshops, authorised to install, activate and calibrate digital and smart tachographs.

ADA\_030 Such approved workshop installing the adaptor shall adjust the input interface and select the division ratio of the input signal (if applicable).

ADA\_031 Such approved workshop installing the adaptor shall seal the adaptor housing.

ADA\_032 The adaptor shall be fitted as close as possible to that part of the vehicle which provides its incoming pulses.

ADA\_033 The cables for providing the adaptor power supply shall be red (positive supply) and black (ground).

## 5.2. Sealing

ADA\_034 The following sealing requirements shall apply:

- the adaptor housing shall be sealed (see ADA\_017),
- the housing of the embedded sensor shall be sealed to the adaptor housing, unless it is not possible to remove the sensor from the adaptor housing without breaking the seal(s) of the adaptor housing (see ADA\_018),
- the adaptor housing shall be sealed to the vehicle,
- the connection between the adaptor and the equipment which provides its incoming pulses shall be sealed on both ends (to the extent of what is reasonably possible).

## 6. Checks, inspections and repairs

### 6.1. Periodic inspections

ADA\_035 When an adaptor is used, each periodic inspection (periodic inspections means in compliance with Requirement [409] through to Requirement [413] of Annex 1C) of the recording equipment shall include the following checks:

- that the adaptor carries the appropriate type approval markings,
- that the seals on the adaptor and its connections are intact,
- that the adaptor is installed as indicated on the installation plaque,
- that the adaptor is installed as specified by the adapter and/or vehicle manufacturer,
- that mounting an adaptor is authorised for the inspected vehicle.

ADA\_036 These inspections shall include a calibration and a replacement of all seals, whatever their state.

## 7. Type approval of recording equipment when an adaptor is used

### 7.1. General points

ADA\_037 Recording equipment shall be submitted for type approval complete, with the adaptor [425].

ADA\_038 Any adaptor may be submitted for its own type approval, or for type approval as a component of a recording equipment.

ADA\_039 Such type approval shall include functional tests involving the adaptor. Positive results to each of these tests are stated by an appropriate certificate [426].

### 7.2. Functional certificate

ADA\_040 A functional certificate of an adaptor or of recording equipment including an adaptor shall be delivered to the adaptor manufacturer only after all the following minimum functional tests have been successfully passed.

<i>No</i>	<i>Test</i>	<i>Description</i>	<i>Related requirements</i>
<b>1. Administrative examination</b>			
1.1	Documentation	Correctness of documentation of the adaptor	
<b>2. Visual inspection</b>			
2.1.	Compliance of the adaptor with documentation		
2.2.	Identification / markings of the adaptor		ADA_027, ADA_028
2.3	Materials of the adaptor		[219] to [223] ADA_026
2.4.	Sealing		ADA_017, ADA_018, ADA_034
<b>3. Functional tests</b>			
3.1	Inducing the speed pulses to the embedded motion sensor		ADA_013
3.2	Interfacing and adapting incoming speed pulses		ADA_011, ADA_012
3.3	Motion measurement accuracy		[30] to [35], [217]
<b>4. Environmental tests</b>			
4.1	Manufacturer test results	Results of manufacturer environment tests.	ADA_020, ADA_021, ADA_022, ADA_024
<b>5. EMC</b>			
5.1	radiated emissions and susceptibility	Verify compliance with Directive 2006/28/EC	ADA_024
5.2	Manufacturer test results	Results of manufacturer environment tests.	ADA_024