



Department
of Energy &
Climate Change

Smart Metering Implementation Programme

Great Britain Companion Specification (GBCS)

Version 0.8.1: Release Note
28 November 2014



Release Note

This release note accompanies, but does not form part of GBCS v0.8.1. It describes the principal changes and updates to GBCS v0.8, and the revision history of GBCS.

Summary of main changes

This document is GBCS v0.8.1.

The following changes apply throughout this document:

- alignment with the published versions of the DLMS Green Book 8 and Blue Book 12;
- alignment with the published version of ZSE 1.2a v0.9;
- alignment with SMETS v1.58 and CHTS v1.46;
- provision of additional material which clarifies or amplifies, but does not change, the requirements published in GBCS v0.8; and
- the correction of a number of minor typographic errors.

In addition, the sections of this document listed below incorporate specific changes and updates to GBCS v0.8:

Section	Changes
4.3.1.5: Protection against Replay Mechanisms	Corrections to align to Section 9.2 - Devices require fewer anti-replay counters than stated in GBCS v0.8.
6.2.4: Command authenticity and integrity verification	The sequencing of authentication checks has been made more explicit, and corrected to reflect the required anti-replay counters (above)
7.2.10: Message construction – GBZ payloads	The addition of an optional 'From Date Time' field (setting the earliest state date / time to read from) to cater for two ZCL commands that do not have such a field
7.2.11: Transfer of large remote party messages	Inclusion of explicit provisions for retry when parts of a message are not received (loss of blocks)
7.3.7: Pricing matrices, scripts and registers	Details the required mapping between price and register related attributes on the ESME, and provides a supporting narrative
7.3.9: ESME accounts, credits and charges	Provision of explanatory information relating to ESME accounts, credits and charges
7.3.10: ESME requirements for using Special Days objects	Makes explicit the required mapping between Special Day objects and the calendars with which each is to be associated
7.4: Device requirements - ZSE	Replacement of original material to clarify the normative / informative structure, and correct a number of detailed points. Specifically, the revised material sets out the ZigBee clusters, attributes and commands that shall be supported by Devices in their interactions with other Devices on the same HAN. The mapping includes the requirements mapping back to SMETS, CHTS and the wider GBCS
9.1: Time synchronisation	Minor drafting corrections
9.2.2: Future dated commands for the writing of attributes	Clarification that all future dated times in a single Command have to be the same



Department
of Energy &
Climate Change

9.2.2.7: ESME requirements for activation of future dateable commands	Makes explicit the requirements within an ESME for activation of future dated commands
10: ZSE implementation	Corrections to align with the published ZigBee specification, including that a PPMID may be sleepy, and that certain notification flags must not be actioned by a GSME
10.3.4: GSME command retrieval and TOM requirements	Corrections to the list of TOM Commands
10.4.2.11: Other attributes	Makes explicit where GBCS has specific processing requirements in relation to ZSE attributes which are not detailed in the ZSE specification
13.7.4.1: Use Case Requirements (Pair-wise Authorisation of Devices)	Inclusion of valid Business Originator role(s) for each type of join and unjoin command
16: Event / Alert Codes and related requirements	Changes to renumber mandated alert codes to avoid clashes with pre-existing alert codes in other standards; clarify that non-mandated alerts are Type 1 (do not have payload); and correct alerts associated with restoration of supply after the load limit restoration period has elapsed
18.1.1.1: Message Templates for ZSE commands between ESME and HCALCS	Addition of a section to make explicit how ZSE commands between an ESME and an HCALCS should be constructed
18.2: DLMS Cosem Message Templates	Updated to reflect changes in the Mapping Table
18.2.1.2: Values of the credit_charge_configuration attribute of Account (Class ID 111) objects	Specification of five possible values for the credit_charge_configuration attribute, and an informative explanation of how these are derived (18.2.1.3)
18.2.1.4: Encoding of Billing Calendar start date-time and periodicity	Addition of a section setting out how the Billing Calendar is to be populated in the Command to an ESME
19: Use Cases	Replacement of embedded file of HTML to reflect updated Mapping Table
20: Mapping Table	See note below

All changes to cells in the Mapping Table at Section 20 are highlighted in yellow. These changes arise from the following sources:

- alignment to the published ZigBee and DLMS specifications;
- correction of misalignment between gas and electricity fuel use cases;
- alignment to SMETS changes, particularly relating to ALCS;
- correction of detailed errors identified; and
- removal of now redundant information, for example Section 7.4 (Device Requirements – ZSE) is no longer generated from the Mapping Table so the relevant cells are now blanked.



Department
of Energy &
Climate Change

Revision History

Version	Rev	Date of Issue	Status	Change Summary
0.1		28/3/13	Draft	Version shared internally for ISFT preparation
0.2		12/4/13	Draft	Updated version to add example content
0.3		15/4/13	Draft	Updates to add RBAC
0.4		22/4/13	Draft	Additional security information added
0.5		26/7/13	Draft	Reformatted to reflect detailed security specifications
0.6		30/08/13	Draft	Updated in light of SDAG comments, corrections and completion of previously unpopulated sections
0.7	5	20/12/13	Draft	Alignment to CHTS 1.32 and SMETS 1.3 and the inclusion of protocol mapping tables
0.7	5RF	29/1/14	Draft	Reformat. Technical content unchanged
0.7	6	7/2/14	Draft	See Release Note in that document
0.7	7	13/5/13	Draft	See Release Note in that document
0.8		8/7/14	Draft	See Release Note in that document
0.8.1		28/11/14	Draft	See Release Note above



Department
of Energy &
Climate Change

Smart Metering Implementation Programme

Great Britain Companion Specification (GBCS)

Version 0.8.1
28 November 2014

© Crown copyright 2014

You may re-use this information (not including logos) free of charge in any format or medium, under the terms of the Open Government Licence.

To view this licence, visit www.nationalarchives.gov.uk/doc/open-government-licence/ or write to the Information Policy Team, The National Archives, Kew, London TW9 4DU, or email: psi@nationalarchives.gsi.gov.uk.

This document is also available from our website at www.gov.uk/decc.

Version: GBCS v0.8.1

Issued: 28 November 2014

Enquiries to:

Smart Metering Implementation Programme
Department of Energy & Climate Change
Orchard 3, Lower Ground Floor
1 Victoria Street
London, SW1H 0ET

Telephone: 0300 068 6659
Email: smartmetering@decc.gsi.gov.uk

Documentation Alignment

SMETS and CHTS

All references in this document to the second version of the Smart Metering Equipment Technical Specifications (SMETS) are to Version 1.58¹. All references to the Communications Hub Technical Specifications (CHTS) are to Version 1.46.

Both these documents can be obtained here:

<https://www.gov.uk/government/consultations/smart-metering-equipment-technical-specifications-second-version>

DLMS Green and Blue Books

This document aligns with the published versions of the Green Book (DLMS UA 1000-2 Ed. 8.0) and Blue Book (DLMS UA 1000-1 Ed. 12.0). These documents can be obtained from the DLMS User Association: <http://www.dlms.com>.

ZigBee Smart Energy Profile

All references in this document to the ZigBee Smart Energy (ZSE) Profile Specification relate to 1.2a v0.9 (reference 14-0256 Rev 04: <http://zigbee.org/About/GBCSPartner.aspx>). The following documents are also referenced and are available from the same link:

- ZigBee Cluster Library (ZCL) Specification, reference 07-5123 Rev 04;
- ZigBee OTA Upgrade Cluster Specification, reference 09-5264 Rev 23;
- ZigBee Specification – 05-3474 Rev 20; and
- ZigBee Pro PICS and Stack Profiles – 08-0006 Rev 05.

Please note that ZigBee Smart Energy Profile 1.2a v0.9 may be subject to minor amendment in accordance with the ZigBee development process prior to its publication as a standard (ZigBee Smart Energy Profile 1.2a v1.0).

Smart Energy Code

Where relevant, the contents of this version of GBCS align with the draft version of the Smart Energy Code (SEC) 4, which was issued for consultation in June 2014.

¹ This document also includes the HAN Connected Auxiliary Load Control Switches (HCALCS) Technical Specification, the Prepayment Interface Device (PPMID) Technical Specification (PPMIDTS), and the In Home Display (IHD) Technical Specification (IHDTs)

Table of Contents

1	Introduction - informative.....	7
2	Structure of the GB Companion Specification (GBCS).....	8
2.1	Normative Requirements.....	8
2.2	Structure of the GB Companion Specification (GBCS) and its relationship to other documents - informative.....	8
3	Scope and Terminology.....	10
3.1	Introduction - informative.....	10
3.2	Scope.....	10
3.3	Terminology.....	11
4	Security.....	13
4.1	Introduction – informative.....	13
4.2	Cryptographic Protections applying to all Messages.....	14
4.3	Security for Remote Party Messages.....	14
5	Remote Party Message Construction, Protection and Verification – informative.....	26
5.1	Common Message Structures - informative.....	26
5.2	Common Encryption and Decryption approach - informative.....	26
5.3	Message Categories - informative.....	26
5.4	Common Message Processing steps - informative.....	27
5.5	Common processing stages and requirements for Devices operated through the DCC - informative.....	29
6	Message Categories.....	33
6.1	Introduction - informative.....	33
6.2	Message Category SME.C.....	33
6.3	Message Category SME.C.C.....	37
6.4	Message Category SME.C.NC.....	38
6.5	Message Category SME.C.PPMID-GSME.....	40
6.6	Message Category SME.A.....	42
6.7	Message Category SME.A.C.....	42
6.8	Message Category SME.A.NC.....	43
7	Message structure and DLMS COSEM / ZSE / ASN.1 requirements.....	45
7.1	Introduction - informative.....	45
7.2	Remote Party Message Construction - general.....	45
7.3	Device Requirements – DLMS COSEM.....	63
7.4	Device requirements – ZSE	70
8	Encryption of Attributes in Remote Party Messages.....	73
8.1	Approach - informative.....	73
8.2	Common requirements.....	73
8.3	Key Derivation Inputs.....	74
8.4	AAD, Plaintext and Ciphertext.....	74
8.5	Access to sensitive data – COSEM attribute access.....	75
8.6	Access to sensitive data – ZSE attribute access.....	81
9	Time Synchronisation and Future Dated Remote Party Messages.....	82
9.1	Time synchronisation.....	82
9.2	Future Dated Remote Party Messages.....	89

10 ZSE Implementation.....	94
10.1 Introduction - informative.....	94
10.2 Tunnels.....	94
10.3 GSME and GPF interactions.....	98
10.4 GPF Structured Data Items.....	101
10.5 Hand Held Terminal (HHT) interactions.....	107
11 Downloading firmware images to Devices.....	112
11.1 Introduction – informative.....	112
11.2 Common Requirements.....	112
11.3 CS05a Distribute Firmware to Communications Hub.....	115
11.4 CS05b Distribute Firmware to ESME / GSME.....	116
11.5 CS06 Activate Firmware.....	117
12 Requirements for Certificates.....	121
12.1 Requirements applicable to all Certificates.....	121
12.2 Requirements applicable to Organisations' Certificates only.....	122
12.3 Requirements applicable to Certificates where <code>RemotePartyRole = root or issuingAuthority</code>	122
12.4 Requirements applicable to Certificates where <code>RemotePartyRole is neither root nor issuingAuthority</code>	123
12.5 Requirements applicable to Device Certificates.....	123
12.6 Device processing of Certificates.....	123
13 Managing Security Credentials on Devices.....	124
13.1 Introduction - informative.....	124
13.2 CS02a Provide Security Credential Details Command and Response.....	126
13.3 CS02b Update Security Credentials Command, Response and Alert.....	138
13.4 CS02c Issue Security Credentials.....	172
13.5 CS02d Update Device Certificates on Device.....	177
13.6 CS02e Provide Device Certificates from Device.....	182
13.7 Pair-wise Authorisation of Devices.....	187
13.8 GCS59 / 62 GPF Device Log Backup and Restore.....	202
14 Apply Prepayment Top Up to an ESME or GSME.....	209
14.1 Defined Terms.....	209
14.2 Description - informative.....	209
14.3 Common Requirements.....	210
14.4 CS01a Applying a Prepayment Top Up to an ESME without consumer intervention	
212	
14.5 CS01b Applying a Prepayment Top Up to a GSME without consumer intervention	
214	
14.6 Applying a Prepayment Top Up to an ESME or GSME with consumer entry of a numeric code on the ESME or GSME.....	215
14.7 Applying a Prepayment Top Up to an ESME or GSME with consumer entry of a numeric code on a PPMID.....	218
14.8 Calculating and Verifying the UTRN Check Digit.....	220
15 Message Codes.....	222
16 Event / Alert Codes and related requirements.....	223
16.1 Introduction – informative.....	223
16.2 Event and Alert Codes.....	224
16.3 Event Logs.....	224

16.4 Requirements.....	224
17 Remote Party Usage Rights.....	226
17.1 Remote Party Access Rights to Attributes and Methods.....	226
17.2 Remote Party Usage Rights to Use Cases.....	226
18 Message Templates.....	227
18.1 GBZ and ZSE Message Templates.....	227
18.2 DLMS COSEM Message Templates.....	230
18.3 Illustrative command and response instantiation and DER encoding.....	251
18.4 Cryptographic Test Vectors.....	260
19 Use Cases.....	279
19.1 Use Case Title.....	279
19.2 Use Case-specific content.....	281
19.3 Embedded Use Cases.....	282
20 Mapping Table.....	283
21 Glossary.....	284
22 Annex 1 – Additional DLMS Class.....	298
22.1 Attribute description.....	298
22.2 Method description.....	299
23 Annex 2 - Counters and their use in transaction identification and Protection Against Replay protection - informative.....	301
24 Annex 3 – ASN.1 modules - informative.....	305
25 Annex 4 - Use of ZigBee in GBCS - informative.....	326
25.1 Purpose.....	326
25.2 GBCS requirements to use ZigBee.....	326
25.3 GBCS requirements not to use ZigBee / vary from it.....	326
26 Annex 5 - Use of DLMS COSEM in GBCS - informative.....	327
26.1 Purpose.....	327
26.2 GBCS requirements to use DLMS COSEM.....	327
26.3 GBCS requirements not to use DLMS COSEM / vary from it.....	328
27 Annex 6 - Deducing the UTRN Counter from the Truncated UTRN Counter – informative.....	329

1 Introduction - informative

- The second version of the Smart Metering Equipment Technical Specifications (SMETS2) requires that Gas Smart Metering Equipment (GSME), and Electricity Smart Metering Equipment (ESME) including variants, meet the requirements described in this Great Britain Companion Specification (GBCS).
- The Communications Hub Technical Specifications (CHTS) requires that Communications Hubs meet the requirements described in this GBCS.
- The HAN Connected Auxiliary Load Control Switches (HCALCS) Technical Specification (HCALCSTS) requires that HCALCS meet the requirements described in this GBCS.
- The Prepayment Interface Device (PPMID) Technical Specifications (PPMIDTS) requires that PPMIDs meet the requirements described in this GBCS.
- GBCS v0.8 was notified to the European Commission in accordance with the requirements of Article 8 of Directive 98/34/EC of the European Parliament and of the Council laying down a procedure for the provision of information in the field of technical standards and regulations (OJ L 204, 21.7.1998, p. 37) as amended by Directive 98/48/EC of the European Parliament and of the Council (OJ L 217, 5.8.1998, p. 18). The Government is currently considering if renotification is required due to the changes made in this version (v0.8.1), compared to v0.8.

2 Structure of the GB Companion Specification (GBCS)

2.1 Normative Requirements

Some sections of the GBCS are informative and others normative. Unless sections are marked 'informative' in the header, they shall be normative. Subsections of sections marked informative shall also be informative.

For defined terms (those capitalised), please see the Glossary at Section 21. Where terms are in *courier new* font, they are Abstract Syntax Notation One (ASN.1²) specified structures defined in this document, or in IETF RFC 5912³. Definitions of such ASN.1 structures are not repeated in the Glossary.

2.2 Structure of the GB Companion Specification (GBCS) and its relationship to other documents - informative

The whole of this Section 2.2 is informative. A number of documents specify what Devices should do and how they should do it, including:

- the Device Specifications (SMETS (including the IHDTs, HCALCSTS and PPMIDTS), and CHTS). These documents:
 - lay out minimum physical requirements and minimum functional capabilities for Devices;
 - specify that all Devices must use the ZSE protocol specifications; and
 - specify that Electricity Smart Metering Equipment (ESME) must additionally use DLMS COSEM protocol specifications.
- International Standards documents, including those which lay out what is required to use ZSE and DLMS COSEM protocols. However, the standards are flexible and could be used in many different ways to implement technically the minimum functional requirements of SMETS and CHTS;
- the end to end protocol that is defined in the GBCS deviates from the standard ZigBee SEP1.2 and DLMS COSEM protocols in some instances. Suppliers and the DCC are required to deploy Devices that are certified against those aspects of the GBCS that are fully compliant with the ZigBee and DLMS COSEM protocols. Certification is not required against those aspects of the GBCS where the ZigBee and DLMS COSEM protocols are actively dis-applied or modified.

For additional information on the level of the area that would not require certification please see Section 25 for ZigBee SEP1.2, and Section 26 for DLMS COSEM.

GB Smart Metering requires technical interoperability, and so requires a single, consistent, technical implementation of the capabilities laid out in SMETS and CHTS across all Devices, in so far as the network communications with Devices are concerned, be those communications over the Smart Metering Home Area Network (SMHAN) or Wide Area Network (WAN). **The Devices in scope of this GBCS are:**

- **Electricity Smart Metering Equipment (ESME)**, including Polyphase, Twin Element, Auxiliary Load Control Switch (ALCS) and Boost Function variants thereof;

² <http://www.itu.int/rec/T-REC-X.680-X.693-200811-I/en>

³ <http://tools.ietf.org/html/rfc5912>

- 58 • Gas Smart Metering Equipment (GSME);
59 • Communications Hub (Communications Hub Function - CHF) and Communications Hub
60 (Gas Proxy Function - GPF);
61 • Prepayment Interface Device (PPMID) and HAN Connected Auxiliary Load Control
62 Switch (HCALCS); and
63 • Type 2 Devices, including In Home Displays (IHDs).

64 The purpose of this GBCS, and related documents, is to specify the single, consistent
65 technical implementation in sufficient detail to achieve operational interoperability of Devices.

66 The Smart Metering technical and security architecture is based on a suite of agreed, open
67 standards, reflecting the UK Government strategy to facilitate the development of third party
68 innovative solutions for consumer devices. These include standards relating to DLMS
69 COSEM, ZSE, ASN.1, NSA Suite B cryptography and X.509 related IETF RFCs. The GBCS
70 does not duplicate what is laid out in such standards but rather provides references to them.

71 3 Scope and Terminology

72 3.1 Introduction - informative

73 This Section 3.1 is informative and summarises Section 3.

74 **Section 3 introduces key terms used in the GBCS:**

- 75 • **Messages are how Devices communicate between themselves and with organisations**
76 **remote from Consumers' Premises.** Such **Messages are 'end-to-end' and 'unicast'** in
77 **that:**
 - 78 ○ **they all identify the sender** (e.g. a Supplier) and the intended recipient (e.g. an
79 ESME); and
 - 80 ○ **they are all intended for processing by the intended recipient**, even though they may
81 pass through intermediate Devices, such as a Communications Hub. Most
82 Messages pass through Communications Hubs unaltered, save for any 'wrapping'
83 information needed for transport purposes. The only exception is where a
84 Communications Hub Device is the intended recipient or is the sender (in these
85 cases the Message is processed by the CHF or GPF), or where covered by the
86 Tapping Off Mechanism (Section 10);
- 87 • **Messages are one of:**
 - 88 ○ a Command to a Device or a corresponding Response;
 - 89 ○ an Alert from a Device; or
 - 90 ○ an information provision transaction (HAN Only Message) solely between Devices;
- 91 • **Organisations (such as Suppliers and Network Operators) communicating with Devices**
92 **are called Remote Parties;**
- 93 • **Messages to and from Remote Parties are called Remote Party Messages;** and
- 94 • Messages solely between Devices are called HAN Only Messages.

95 Section 3 then:

- 96 • explains that the GBCS only covers the Messages needed for the minimum functionality
97 laid out in the SMETS and CHTS;
- 98 • explains that the GBCS specifies how all such Messages are constructed and related
99 processing performed; and
- 100 • notes that Type 2 Devices (e.g. IHDs) can only send or receive HAN Only Messages.

101 Section 3 also explains some technical terminology and technical conventions used in this
102 GBCS.

103 3.2 Scope

104 This Section 3.2 lays out the scope of the GBCS and introduces definitions relied upon in
105 this GBCS.

106 **A Message shall be of one the following:**

- 107 • a Command;
- 108 • a Response to a Command;
- 109 • an Alert; or

- 110 • an information provision transaction (HAN Only Message).

111 A Message instance shall be an instance of one of the Messages detailed in this GBCS.

112 The Device Specifications define the minimum functional capabilities required of Devices.

113 Except where those functional capabilities are internal to the Devices or are accessed via
114 the Device's User Interfaces, the minimum functional capabilities shall be invoked by, and /
115 or result in, Messages being passed via the Devices' Network Interfaces.

116 The GBCS is the technical specification, sufficient for the creation by the originator(s) and
117 processing by the target(s), of each Message, where the Message is required in order to
118 implement minimum functionality defined in the Device Specifications.

119 Specifically, the GBCS details the format, structure and associated processing for each of
120 the Messages required to implement the Device Specifications' minimum functionality.

121 There are two classifications of Message:

- 122 • HAN Only Message⁴, where both the original sender and ultimate recipient are Devices
123 within the same Smart Metering Home Area Network (SMHAN); and
- 124 • Remote Party Message, where either the original sender or the ultimate recipient is not
125 a Device.

126 A Remote Party Message shall only be of one of the following:

- 127 • a Command;
- 128 • a Response to a Command; or
- 129 • an Alert.

130 Each Remote Party Message shall have a unique Message Code, which shall be as
131 specified in Section 15.

132 Where a Remote Party is known to a Device by way of that Remote Party's Security
133 Credentials being stored on the Device (as specified in Section 4.3.2.5), the Remote Party is
134 referred to as a Known Remote Party (KRP). Otherwise, it is referred to as an Unknown
135 Remote Party (URP).

136 Commands requiring a Response to an Unknown Remote Party shall always be sent to the
137 Device by the Device's Access Control Broker (see Section 4.3.2.5).

138 For clarity, Type 2 Devices shall not be required to support any Remote Party Messages.
139 Thus, provisions in this GBCS in relation to Remote Party Messages shall not apply to Type
140 2 Devices.

141 Remote Parties and Devices are collectively referred to in this GBCS as Smart Metering
142 Entities.

143 3.3 Terminology

144 3.3.1 Numbers

145 Numbers within this GBCS are expressed in one of three ways, to avoid potential ambiguity:

- 146 • where a number has no prefix, it is a decimal number (base 10);
- 147 • the 0x prefix is used for hexadecimal numbers (base 16). For example, 0x10 equates to
148 the decimal number 16; and

⁴ HAN Only Messages are ZigBee commands or response commands. This includes HAN Only Messages passed between Devices using the ZSE TransferData, for example a Command from a PPMID to a GSME.

- 149 • the 0b prefix is for binary numbers (base 2). For example, 0b1010 equates to the
 150 decimal number 10.

151 3.3.2 Bit numbering

152 Numbering of bits uses the ‘LSB 0’ bit numbering scheme, where the least significant bit is
 153 referred to as bit 0 and the most significant bit is referred to using the highest bit number.

154 3.3.3 Octets and bytes - informative

155 The term ‘octet’ is used to refer to units of 8 bits of digital information, to avoid potential
 156 ambiguity with the term ‘byte’, and to align with protocol terminology.

157 3.3.4 Tag and MAC - informative

158 In this GBCS:

- 159 • the word ‘tag’ is always used in the sense it is meant in encoding standards, such as A-
 160 XDR⁵ and Distinguished Encoding Rules (DER)⁶;
- 161 • ‘tag’ is never used to mean Authentication tag, in the cryptographic sense;
- 162 • ‘MAC’ is always used to mean Message Authentication Code, which is a cryptographic
 163 checksum on data. Thus, MAC is used instead of Authentication tag; and
- 164 • ‘MAC’ is never used to refer to Medium Access Control, as used in ‘MAC address’,
 165 which is a unique identifier assigned to network interfaces.

166 3.3.5 Concatenation

167 $X \parallel Y$ shall mean the concatenation of the two octet strings X and Y .

168 For example:

169 $X = 0xCAFE$
 170 $Y = 0xBEEF$
 171 $X \parallel Y = 0xCAFEBEEF$

172 3.3.6 Encoding and length of variable length unsigned integers

173 Encoding(X) shall be the encoding of a variable size unsigned integer X as follows:

- 174 • if $0 < X < 128$, then Encoding(X) is a single octet whose value is X ; or
- 175 • if $128 \leq X < 32,768$, then Encoding(X) is a an octet string composed of the
 176 concatenation $0x82 \parallel Y$, where Y is two octets in length and has a value equal to the
 177 two’s complement representation of the value X ; or
- 178 • if $32,768 \leq X < 8,388,608$, then Encoding(X) is a an octet string composed of the
 179 concatenation $0x83 \parallel Y$, where Y is three octets in length and has a value equal to the
 180 two’s complement representation of the value X .

181 Len(Encoding(X)) shall be the length in octets of Encoding(X), so shall be either 1 ($X < 128$),
 182 3 ($128 \leq X < 32768$) or 4 ($32,768 \leq X < 8,388,608$).

183 3.3.7 GeneralizedTime

184 The GeneralizedTime ASN.1 type used in this GBCS shall be a UTC Time with a
 185 resolution of one second. See Section 46 of the ASN.1 specification for format.

⁵ IEC 61334-6

⁶ <http://www.itu.int/ITU-T/studygroups/com17/languages/X.690-0207.pdf>

4 Security

4.1 Introduction – informative

This Section 4.1 is informative and summarises Section 4.

Section 4.2 lays out security provisions that are common across Messages, specifically stating that:

- at the application layer, all Messages must have integrity and authenticity protections, Critical Messages must have non-repudiation protections and some parts of Messages must have Confidentiality protections applied to specific data content; and
- ZSE protections will be relied upon when Devices within the same Smart Metering Home Area Network (SMHAN) communicate with each other.

Section 4.3 lays out security provisions that are common across Remote Party Messages, specifically:

- *Identifiers, Counters and Protection Against Replay*: lays out requirements in relation to identifiers, counters and their use in Protection Against Replay;
- *Security Credentials*: lays out requirements for all Devices, except for Type 2 Devices, to:
 - have Public-Private Key Pairs, and to make their Public Keys available; and
 - have Trust Anchor Cells, including those which are storage areas within a Device, capable of holding Public Key Security Credentials for a number of Remote Parties, with the set of Remote Parties being derived from the functionality the Device supports; and
- *Cryptographic Primitives and their Usage*: lays out requirements for Cryptographic Algorithms and their usage, in relation to Remote Party Messages.

Note that the cryptographic protections are intentionally independent of whether a Message Payload is structured according to the ZSE, ASN.1 or DLMS COSEM standards. This means that Suppliers, Network Operators, the Access Control Broker and Other Users who may communicate with Devices need only implement cryptographic requirements in one way, regardless of the type of Device they are communicating with.

The same requirements for security apply regardless of whether a Message is delivered by the Wide Area Network (WAN), SMHAN, Hand Held Terminal (HHT) or local interface. Note that, for Prepayment Top Up, there are a number of different Messages. The content of each particular Message will always be processed in the same way regardless of delivery mechanism. The governance and structures to ensure uniqueness of identifiers are set out in the Smart Energy Code (SEC) and SMETS, and are outside the scope of the GBCS.

A single Originator Counter can be used for the whole of a Remote Party Organisation (e.g. by that Party counting small enough time intervals). A separate counter per Device is not required.

The Supplementary Originator Counter as specified in Section 4.3.1.4 is required where the corresponding Response has to be cryptographically protected (by way of Encryption, a MAC, or both), to the Supplementary Remote Party. In all other cases, the Response is protected back to the Access Control Broker.

Smart Metering entities make extensive use of a range of Counters as part of the unique identification of Smart Metering Messages. Counters are also a key component used to

229 support Protection Against Replay functionality. An overview of each of these counters and
 230 their use is included as Section 23.

231 **4.2 Cryptographic Protections applying to all Messages**

232 Each Message shall have Cryptographic Protections to give assurance to the Message
 233 recipient(s) as to:

- 234 • the Message's integrity; and
- 235 • the Authenticity of the party or parties creating or augmenting the Message.

236 The minimum set of such Cryptographic Protections is laid out in this GBCS.

237 This GBCS lays out the Cryptographic Protections for non-repudiation, where this quality is
 238 required for specific Messages, so for Critical Messages.

239 Where part of a Message is Confidential, that part shall have Cryptographic Protections to
 240 ensure both its Confidentiality and its integrity, as detailed in this GBCS.

241 For HAN Only Messages the Cryptographic Protections required by this GBCS shall be
 242 those provided by ZSE.

243 For clarity, the HAN Only Message Cryptographic Protections require that all Devices shall:

- 244 • be provisioned with the corresponding ZSE related Security Credentials; and
- 245 • be capable of performing the associated cryptographic operations.

246 **4.3 Security for Remote Party Messages**

247 This Section 4.3 shall:

- 248 • apply only to Remote Party Messages;
- 249 • apply to all Remote Party Messages, regardless of the mechanism (i.e. across the WAN,
 250 SMHAN, HHT or User Interface) by which they are delivered to, or received from, the
 251 Device in question; and
- 252 • apply to the processing of Remote Party Messages by Remote Parties and Devices.

253 **4.3.1 Identifiers, Counters and Protection Against Replay**

254 ***4.3.1.1 Identifiers***

255 All Smart Metering Entities shall have an Entity Identifier which shall be an octet string of
 256 length 8. Each Entity Identifier shall be unique across GB Smart Metering.

257 Entity Identifiers shall be used in the Business Originator ID and Business Target ID fields of
 258 Remote Party Messages as shown in Table 4.3.1.1.

Message Type	Business Originator ID	Business Target ID
Command	Entity Identifier for the Known Remote Party which is requesting execution of this Command	Entity Identifier for the Device that the Remote Party wants to action the Command
Response	The Entity Identifier for the Device. This is always the same as the Business Target ID supplied in the corresponding Command	The Business Originator ID provided in the corresponding Command For Commands to which the corresponding Response is intended for an Unknown Remote Party, the Business Originator ID in the Command shall

		always be that of the Access Control Broker
Alert	The Entity Identifier for the Device	The Entity Identifier for the Known Remote Party to which the Alert is to be addressed. Section 16 of this GBCS specifies which Known Remote Party role each type of Alert shall be addressed to

Table 4.3.1.1: Entity Identifiers for Business Originator and Target ID fields

4.3.1.2 Originator Counter

Except where specified otherwise in the GBCS, a Remote Party Message shall include an Originator Counter, which shall be octet string of length 8 whose contents shall be set and read as an unsigned 64-bit integer. Responsibility for generating the Originator Counter shall be as shown in Table 4.3.1.2.

Message	Responsibility for generating the Originator Counter
Command	The Known Remote Party identified by the Business Originator ID in the Command.
Response	The Originator Counter shall have the same value as in the corresponding Command.
Alert	The Device generating the Alert.

Where a Device is required to generate an Originator Counter, the Device shall ensure that the value it generates is strictly numerically greater than any previous Originator Counter value it has placed in any previous Message it has generated, and strictly numerically greater than any Supplementary Originator Counter it has placed in any previous Message it has generated.

Where a Remote Party is required to generate an Originator Counter, the Remote Party shall ensure that:

- the value it generates is strictly numerically greater than any previous Originator Counter value it has provided for use in any previous Command to the Device in question;
- the 32 least significant bits shall not all have the value 0b0 unless the Command is a Prepayment Top Up Command (see Section 14.3.6 for use of the Originator Counter as the UTRN Counter); and
- if the Command is a Prepayment Top Up then the 32 least significant bits shall all have the value 0b0.

4.3.1.3 Message Identifier

A Message Identifier shall be the concatenation:

Business Originator ID || Business Target ID || CRA Flag || Originator Counter

All Messages shall include a Message Identifier which shall be:

- constructed according to the requirements of this Section 4.3.1; and
- incorporated in the Message according to the requirements of Section 7.

4.3.1.4 Additional Counters and Identifiers

The following attributes shall be incorporated in Commands where (1) the Business Originator ID is set to be that of the Access Control Broker and (2) the Message Code is listed in the 'Use Case reference' worksheet of the Mapping Table as 'Supplementary Remote Party Data required':

- 290 • Supplementary Remote Party ID, which shall be the Entity Identifier of the Remote Party
291 requesting the creation of the Command by the Access Control Broker; and
 - 292 • Supplementary Remote Party Counter, which shall be an octet string of length 8.
- 293 All Responses to such Commands shall incorporate:
- 294 • the same Supplementary Remote Party ID and Supplementary Remote Party Counter
295 as the Command; and
 - 296 • for those marked as 'Supplementary Originator Counter required in Response' in the
297 'Use Case reference' worksheet of the Mapping Table, a Supplementary Originator
298 Counter which shall be generated by the Device, shall be an octet string of length 8
299 whose contents shall be set and read as an unsigned 64-bit integer. The Device shall
300 ensure that the value it generates is strictly numerically greater than any previous
301 Originator Counter value it has placed in any previous Message it has generated, and
302 strictly numerically greater than any Supplementary Originator Counter it has placed in
303 any previous Message it has generated.

304 **4.3.1.5 Protection Against Replay mechanisms**

305 Where a Device supports one or more Remote Party Commands that are marked as
306 requiring 'Protection Against Replay' in the Use Cases, the Device shall implement the
307 requirements detailed in this Section 4.3.1.5.

308 For each type of Command that a Device supports, and that is marked as requiring
309 'Protection Against Replay' in its Use Case, the Device shall:

- 310 • have the capability to store an Originator Counter value for each Remote Party Role
311 allowed to request execution of that type of Command (the 'Execution Counter'); and
- 312 • have all Execution Counters initially set to zero at manufacture.

313 **4.3.2 Security Credentials**

314 **4.3.2.1 Introduction – informative**

315 A Device shall be able to process four kinds of Security Credential Document:

- 316 • its own Security Credential Documents, provided in the form of Device Certificates.
317 Here the Device needs processing to cover (1) generating new Public-Private Key Pairs
318 and so issuing Device Certificate Signing Requests, (2) storing its Device Certificates
319 and (3) providing a copy of those Device Certificates on request;
- 320 • Security Credential Documents relating to Known Remote Parties, provided in the form
321 of Organisation Certificates. For these, the Device needs to be capable of (1) storing, (2)
322 replacing and (3) providing details of those it holds on request;
- 323 • Security Credential Documents relating to Unknown Remote Parties, provided in the form
324 of Organisation Certificates. For these, the Device will receive them in a
325 Command so that parts of the Response can be Encrypted. The Device does not need
326 to store such Documents; and
- 327 • Security Credential Documents relating to Certification Authorities, provided in the form
328 of Certification Authority Certificates. These are processed by the Device only when
329 replacing Remote Parties' Security Credential Documents.

330 Sections 8 and 13 cover the above functionality.

331 Section 13 covers requirements related to the structure and content of such Security
332 Credential Documents, where such requirements are relevant to Device processing
333 requirements.

334 This Section 4.3.2 covers requirements for the storage of such Security Credentials on
335 Devices and their usage in verifying cryptographic protections on Commands the Device
336 receives.

337 **4.3.2.2 Security Credential Documents**

338 A Security Credential Document shall be either:

- 339 • a Device Certificate; or
- 340 • a Remote Party's Organisation Certificate; or
- 341 • a Certification Authority Certificate.

342 **4.3.2.2.1 Device Certificate**

343 A Device Certificate shall relate to only one Device and shall meet the requirements
344 specified at Section 12. A Device Certificate shall either be used for Key Agreement or
345 Digital Signing but not both. Device Certificates shall only be issued by Authorised Public
346 Key Infrastructure (APKI) issuing Certificate Authorities. Where Security Credentials relating
347 to a Device are incorporated in a Message, the Security Credentials shall be incorporated in
348 the Message in the form of the Device Certificate.

349 **4.3.2.2.2 Remote Party's Certificate**

350 A Remote Party Certificate shall be one of that Remote Party's Organisation Certificates and
351 so shall relate to only one Remote Party and shall meet the requirements specified at
352 Section 12. As per Section 12, except where remotePartyRole = root a Remote Party
353 Certificate shall either be used for Key Agreement or Digital Signing but not both. Remote
354 Party Certificates shall only be issued by APKI authorised issuing Certificate Authorities.
355 Where Security Credentials relating to a Remote Party are incorporated in a Message, the
356 Security Credentials shall be incorporated in the Message in the form of the Remote Party's
357 Certificate.

358 **4.3.2.2.3 Certification Authority Certificate**

359 A Certification Authority Certificate shall relate to only one Certification Authority and shall
360 meet the requirements specified at Section 12. A Certification Authority Certificate shall only
361 be used by a Device for verifying Digital Signatures on Certificates. Where Security
362 Credentials relating to a Certification Authority are incorporated in a Message, the Security
363 Credentials shall be incorporated in the Message in the form of the Certification Authority's
364 Certificate.

365 **4.3.2.3 Device Security Credentials**

366 Where a Device is of deviceType that is gSME, eSME,
367 communicationsHubCommunicationsHubFunction, or
368 communicationsHubGasProxyFunction, that Device shall have the capacity to store
369 and use securely four private keys:

- 370 • for Key Agreement, a Current Private Key and a Pending Private Key; and
- 371 • for Digital Signing, a Current Private Key and a Pending Private Key.

372 Where a Device is of deviceType that is
373 type1HANConnectedAuxiliaryLoadControlSwitch or
374 type1PrepaymentInterfaceDevice, that Device shall have the capacity to store and
375 use securely two private keys:

- 376 • for Key Agreement, a Current Private Key; and
- 377 • for Digital Signing, a Current Private Key.

- 378 These stores shall be referred to as Private Key Cells.
- 379 Wherever one of a Device's Private Keys is required to be used by a GBCS Cryptographic
380 Protection process, only the relevant Current Private Key shall be used. A Device shall not
381 use any Pending Private Key in any GBCS Cryptographic Protection.
- 382 Where a Device holds a Private Key that is to be used for Key Agreement, the
383 corresponding Public-Private Key Pair shall have been generated according to the NSA's
384 'Suite B Implementer's Guide to NIST SP 800-56Ar2' using the 'Key Pair Generation Using
385 Extra Random Bits' method.
- 386 Where a Device holds a Private Key that is to be used for Digital Signing, the corresponding
387 Key Pair shall have been generated according to the NSA's 'Suite B Implementer's Guide to
388 FIPS 186-3 (ECDSA), February 3, 2010'⁷ using the 'ECC Key Pair Generation Using Extra
389 Random Bits' method.
- 390 Where a Device supports the processing of Remote Party Messages, the Device shall:
- 391 • have two Trust Anchor Cells to store two Device Certificates relating to itself, with one
392 Trust Anchor Cell for storing Device Certificates where keyUsage = keyAgreement
393 and one for Device Certificates where keyUsage = digitalSignature;
 - 394 • where those two Trust Anchor Cells are populated, ensure the Device Certificates have
395 the following attributes:
 - 396 ○ both Device Certificates meet the requirements specified at Section 13;
 - 397 ○ both Device Certificates' hwSerialNum fields have a value the same as the Devices'
398 Entity Identifier; and
 - 399 ○ each Device Certificate's keyUsage field has the same value as the Trust Anchor
400 Cell in which it is placed.

4.3.2.4 Remote Party Security Credentials

- 402 A Device shall only action a Remote Party Command where:
- 403 • the Known Remote Party identified by the Command has, according to the Security
404 Credentials held on the Device, a Remote Party Role which, according to the Mapping
405 Table for the Message Code in question, is allowed to request execution of the
406 Command; and
 - 407 • the Cryptographic Protections in the Command instance received by the Device have
408 been verified, in line with the requirements for a Command with the Message Code in
409 question.
- 410 To enable this, Security Credentials relating to the Remote Parties in question:
- 411 • shall be held in Trust Anchor Cells on the Device; and
 - 412 • shall act as the corresponding Trust Anchors.

4.3.2.5 Required Trust Anchor Cells and related Device requirements

414 The Trust Anchor Cells specified in Table 4.3.2.5 by TrustAnchorCellIdentifier are
415 those required on each deviceType. Additionally:

- 416 • a GSME shall have a Trust Anchor Cell capable of storing Key Agreement Security
417 Credentials for a PPMID; and
- 418 • a PPMID shall have a Trust Anchor Cell capable of storing Key Agreement Security
419 Credentials for a GSME.

⁷ <http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf>

420 The types of Device and the corresponding value of deviceType shall be defined in ASN.1
 421 notation by:

```
422 DeviceType ::= INTEGER {
423   gSME                               (0),
424   eSME                               (1),
425   communicationsHubCommunicationsHubFunction (2),
426   CommunicationsHubGasProxyFunction      (3),
427   type1HANConnectedAuxiliaryLoadControlSwitch (4),
428   type1PrepaymentInterfaceDevice        (5),
429   type2                               (6)
430 }
```

431 Every Device shall:

- 432 have storage allocated capable of holding Security Credentials as required by Table 4.3.2.5 for its Device type; and
- 434 have all the Trust Anchor Cells, specified in Table 4.3.2.5 as being required for its Device type, populated with Security Credentials that comply with the requirements of this GBCS. Critically, root, recovery and accessControlBroker Trust Anchor Cells shall be populated with valid credentials for each of those three Remote Parties.

			Type of Device (✓ = is required; empty = is not required)					
			ESME	GSME	CH (CHF)	CH (GPF ⁸)	HCALCS	PPMID
deviceType value(s)			1	0	2	3	4	5
TrustAnchorCellIdentifier								
No	remotePartyRole	keyUsage	cellUsage					
1	Root	keyCertSign	Management	✓	✓	✓	✓	✓
2	Recovery	digitalSignature	Management	✓	✓	✓	✓	✓
3	Supplier	digitalSignature	Management	✓	✓		✓	✓
4	Supplier	keyAgreement	Management	✓	✓		✓	
5	Supplier	keyAgreement	prePaymentTopUp	✓	✓			
6	networkOperator	digitalSignature	Management	✓			✓	
7	networkOperator	keyAgreement	Management	✓			✓	
8	accessControlBroker	digitalSignature	Management			✓		✓
9	accessControlBroker	keyAgreement	Management	✓	✓	✓	✓	✓

⁸ Supplier and Network Operator credentials on the Communications Hub (Gas Proxy) relate to the supply of gas only. These Trust Anchor Cells on a Communications Hub are still required and valid where there is no GSME connected to the SMHAN, but the stores should be populated with Access Control Broker certificates (so ensuring the Gas Proxy functionality, apart from Update Security Credentials, is inoperable)

10	transitionalCos	digitalSignature	Management	✓	✓		✓	✓	
11	wanProvider	digitalSignature	Management			✓			

Table 4.3.2.5: Requirements for Trust Anchor Cells by Device Type

For clarity, the GPF and CHF shall each have their own set of Trust Anchor Cells.

A specific Trust Anchor Cell shall be identified in this GBCS using the notation {remotePartyRole, keyUsage, cellUsage}. For example {supplier, digitalSignature, management} shall refer to the Trust Anchor Cell that holds the Device's Supplier Digital Signing Security Credentials, so including the Supplier's:

- Entity Identifier;
- Remote Party Role; and
- Digital Signing Public Key.

Where a Device supports the processing of Remote Party Messages, that Device:

- shall support the processing of the Update Security Credentials Command; and
- shall not allow execution of any Remote Party Command other than an Update Security Credentials Command or a Provide Security Credentials Command, nor issue any Remote Party Alerts, in relation to a Remote Party Role where the Remote Party Role stored in a Trust Anchor Cell is different than that of the Trust Anchor Cell itself.

When verifying a Cryptographic Protection applied to a Command instance it receives, a Device shall use the Remote Party Security Credentials that it holds at the time of Command processing.

Devices shall only be capable of replacing Remote Party Security Credentials on receipt of an Update Security Credentials Command specified in this GBCS.

4.3.2.6 What is the Public Key in each Trust Anchor Cell to be used for – informative

TrustAnchorCellIdentifier			Usage of the Public Key in the Trust Anchor Cell
remotePartyRole	keyUsage	cellUsage	
Root	keyCertSign	management	Used only in Certification Path Validation to check that Certification Authority Certificates and Certificates related to change of root credentials were validly issued
Recovery	digitalSignature	management	Used only to verify recovery's signature on Update Security Credentials Commands addressed to the Device
Supplier	digitalSignature	management	Used to verify the supplier's signature on Critical Commands the supplier has addressed to the Device
Supplier	keyAgreement	management	Used in applying MACs to Alerts and Responses addressed to the supplier, where they are not Critical Used in unencrypting encrypted data in Commands from the supplier and in encrypting data

TrustAnchorCellIdentifier			Usage of the Public Key in the Trust Anchor Cell
remotePartyRole	keyUsage	cellUsage	
			in Alerts and Responses addressed to the supplier
Supplier	keyAgreement	prePaymentTopUp	Used to check the supplier MAC on prepayment top up Commands. The supplier can decide whether this is the same key as the Key Agreement key used for other purposes
networkOperator	digitalSignature	management	Used to check the signature of the networkOperator on Critical Commands the networkOperator has sent to the Device. This only equates to Update Security Credentials Commands
networkOperator	keyAgreement	management	Used in applying MACs to Alerts and Responses addressed to the networkOperator, where they are not Critical Used in encrypting data in Responses addressed to the networkOperator
accessControlBroker	digitalSignature	management	Used to verify the accessControlBroker's signature on Commands addressed to the Device
accessControlBroker	keyAgreement	management	Used in checking the accessControlBroker MAC on Commands received and to calculate the MAC for Responses addressed to the accessControlBroker
transitionalCoS	digitalSignature	management	Used only to check transitionalCoS's signature on Update Security Credentials Commands received by the Device
wanProvider	digitalSignature	management	Used by the Communications Hub (CHF) to verify the wanProvider's signature on Critical Commands addressed to the Communications Hub

459 Table 4.3.2.6: Use of Public Keys in each Trust Anchor Cell

460 **4.3.2.7 Mapping a Command to the Remote Party Security Credentials to be used in 461 verifying the Command's cryptographic protections**

462 Except for the Security Credentials related Commands (see Section 13), a Device shall 463 apply the requirements of this Section 4.3.2.7 to identify which of the Remote Party Public 464 Keys that it holds are to be used to verify the cryptographic protections on a Command.

465 **4.3.2.7.1 Message Authentication Codes**

466 Where a Command is a Prepayment Top Up Command, the supplier MAC in that 467 Command shall be verified using the Public Key in Trust Anchor Cell {remotePartyRole}

468 supplier, **keyUsage** keyAgreement, **cellUsage** prePaymentTopUp}, along with
 469 the Device's Key Agreement Private Key.

470 All other MACs in Commands shall be verified using the Public Key in Trust Anchor Cell
 471 {**remotePartyRole** accessControlBroker, **keyUsage** keyAgreement,
 472 **cellUsage** management}, along with the Device's Key Agreement Private Key.

4.3.2.7.2 Signature

474 Where a Command has a Digital Signature, the Device shall identify the Remote Party
 475 Role(s) which can legitimately sign the Command according to the message code identified
 476 in the Mapping Table.

477 If there is only one Remote Party Role so identified, then the signature shall be verified using
 478 the Public Key in Trust Anchor Cell {**remotePartyRole** (the identified remote party role),
 479 **keyUsage** digitalSignature, **cellUsage** management}.

480 If there is more than one Remote Party Role so identified, the Device shall use the Business
 481 Originator ID in the Command to identify the Trust Anchor Cell(s) where:

- 482 • **keyUsage** = digitalSignature;
- 483 • **cellUsage** = management; and
- 484 • **existingSubjectUniqueID** = the Business Originator ID in the Command

485 If there is only one Trust Anchor Cell so identified, then the signature shall be verified using
 486 the Public Key in that Trust Anchor Cell.

487 If there is more than one Trust Anchor Cell so identified the Device shall attempt to verify the
 488 Digital Signature using each Trust Anchor Cell identified. These attempts shall be
 489 according to the following precedence, and attempts to verify shall cease when a signature
 490 verification succeeds:

- 491 1. supplier
- 492 2. wanProvider
- 493 3. networkOperator
- 494 4. accessControlBroker

495 For clarity, other Remote Party Roles on Devices are limited to Commands related to
 496 Security Credentials and so cannot have Trust Anchor Cells identified according to this
 497 Section 4.3.2.7.2.

4.3.2.8 Certification Path Validation

4.3.2.8.1 Access Control Broker requirements

500 Before it calculates the Access Control Broker to Device MAC (ACB-SMD MAC) in line with
 501 Section 6.2.3, the Access Control Broker shall undertake Certification Revocation List (CRL)
 502 Validation for any Organisation Certificate in a Command:

- 503 • either by using the algorithm specified in IETF RFC 5280⁹ Section 6.3; or
- 504 • by using functionality equivalent to the external behaviour resulting from that algorithm.

505 Only if the CRL Validation is successful shall the Access Control Broker calculate the ACB-
 506 SMD MAC. For clarity, the Access Control Broker shall never send a Message to a Device
 507 which contains any Certificate that has failed CRL Validation.

⁹ <http://datatracker.ietf.org/doc/rfc5280/>

508 **4.3.2.8.2 Device requirements**

509 The requirements in this Section 4.3.2.8.2 shall apply only to Use Case CS02b (Update
510 Security Credentials).

511 Where a Device has successfully completed all required Command Authenticity and Integrity
512 checks on a Command of type covered by Use Case CS02b it has received, the Device shall
513 undertake either:

- 514 • Certification Path Validation, including time checks; or
- 515 • Certification Path Validation, excluding time checks.

516 If the Device does not have Reliable Time (as defined in Use Cases GCS28 and ECS70 Set
517 Clock) it shall always undertake Certification Path Validation, excluding time checks.
518 Otherwise the validation to be undertaken shall be determined by the contents of the
519 Remote Party Command instance. For clarity, Device types which are not required to have
520 a clock, shall always undertake Certification Path Validation, excluding time checks.

521 The Device shall undertake Certification Path Validation, including time checks:

- 522 • either by using the algorithm specified in IETF RFC 5280 Section 6.1; or
- 523 • by using functionality equivalent to the external behaviour resulting from that algorithm.

524 The Device shall undertake Certification Path Validation, excluding time checks:

- 525 • either by using the algorithm specified in IETF RFC 5280 Section 6.1 but not applying
526 the check at 6.1.3 (a) (2) ('the certificate validity period includes the current time'); or
- 527 • by using functionality equivalent to the external behaviour resulting from that algorithm
528 where not applying the check that 'the certificate validity period includes the current
529 time'.

530 The 'trust anchor' information (with the meaning in IETF RFC 5280) shall be in the root
531 Security Credentials held on the Device.

532 If the Device's Certificate Path Validation does not confirm the required certification path
533 validity, then the Device shall undertake no further processing of the Command, except for
534 the issuance of a Response notifying that the Command was unsuccessful.

535 **4.3.2.9 DLMS Client and Server**

536 The Access Control Broker shall perform the role of DLMS COSEM client in relation to the
537 DLMS COSEM Application Associations, and the Device shall perform the role of DLMS
538 COSEM server.

539 **4.3.3 Cryptographic Primitives and their Usage**

540 In relation to any Remote Party Message, Smart Metering Entities shall:

- 541 • use SHA-256, as specified in FIPS 180-4¹⁰, as the Hash function;
- 542 • use the AES-128 cipher, as specified in FIPS 197¹¹, as the block cipher primitive;
- 543 • use the Galois Counter Mode (GCM) mode of operation as specified in NIST Special
544 Publication 800-38D¹²;
- 545 • use the GMAC technique, based on the use of AES-128, for the calculation of Message
546 Authentication Codes (MACs), as specified in NIST Special Publication 800-38D (see
547 above);

¹⁰ <http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf>

¹¹ <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>

¹² <http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf>

- 548 • use, as the Digital Signature technique, ECDSA (as specified in *FIPS PUB 186-4*¹³) in
549 combination with the curve P-256 (as specified in *FIPS PUB 186-4* at Section D.1.2.3)
550 and SHA-256 as the Hash function. Within Messages, Signatures shall be in the Plain
551 Format;
- 552 • use, to calculate the Shared Secret Z, the Static Unified Model, C(0e, 2s, ECC CDH)
553 Key Agreement technique (as specified in *NIST Special Publication 800-56Ar2*¹⁴ save
554 for the requirement to zeroize the Shared Secret) with:
 - 555 ○ the Single-step Key Derivation Function (KDF) based on SHA-256, as specified in
556 *NIST Special Publication 800-56Ar2*; and
 - 557 ○ the P-256 curve for the elliptic curve operations.

558 Resulting DerivedKeyingMaterial (with its meaning in *NIST Special Publication 800-56Ar2*)
559 shall only ever be used in relation to one Message instance. Any Shared Secret that is not
560 'zeroized' shall be stored and used with the same security protections as Private Keys.

561 **4.3.3.1 Scope of Cryptographic Protections**

562 The fields that shall always contribute to MAC and Digital Signature are detailed in Section
563 7.2. Fields that vary across Messages are specified in Section 6, and in the relevant Use
564 Cases. For clarity, a Message instance may transit through multiple Smart Metering Entities
565 before delivery to its target Device, and more than one Smart Metering Entity may be
566 required to apply a Cryptographic Protection to that Message instance. Thus, the scope of
567 protection can only be across fields in the Message instance as constructed at the point the
568 protection is applied.

569 Where a Message has multiple Cryptographic Protections, the order in which the Smart
570 Metering Entities apply these Cryptographic Protections is specified in this GBCS.

571 A Device verifying the Cryptographic Protections in such Messages shall undertake such
572 verifications in the reverse sequence to that in which the Cryptographic Protections were
573 applied. This order is also specified in this GBCS.

574 **4.3.3.2 ECDSA per message secret number**

575 When generating a Digital Signature, the Smart Metering Entity shall calculate the DSA Per-
576 Message Secret Number 'k' with respect to ECDSA (with the meaning in Section 4.5 of *FIPS*
577 186-4) to be the SHA-256 hash of the concatenation of:

- 578 • the parts of the Message to be signed, as defined in Section 7.2.7; and
- 579 • the Private Key that the Smart Metering Entity will use in the Digital Signature
580 generation.

581 If the value of k so calculated results in an 'r' or 's' value of 0, where r and s have the
582 meanings in the NSA's 'Suite B Implementor's Guide to FIPS 186-3', then a new value for k
583 shall be calculated to be the SHA-256 hash of the concatenation of:

- 584 • the parts of the Message to be signed, as defined in Section 7.2.7;
- 585 • the Private Key that the Smart Metering Entity will use in the Digital Signature
586 generation; and
- 587 • 0x00.

588 The addition of 0x00 to the concatenation shall be repeated until a value of k is generated
589 that does not result in an 'r' or 's' value of 0.

¹³ <http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf>

¹⁴ <http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar2.pdf>

590 **4.3.3.3 Calculating unique Shared Secret Keys for a Remote Party Message Instance**

591 Where a Smart Metering Entity executes the KDF in relation to a Message instance, the
 592 *OtherInfo* field, with the meaning in *NIST Special Publication 800-56Ar2*, shall be populated
 593 using the value of information provided in, or to be placed in, the originator-system-title,
 594 recipient-system-title and transaction-id fields of the Grouping Header, as per the
 595 requirements of Section 7.2.7.

596 The *OtherInfo* shall be in the Concatenation Format as defined in Section 5.8.1.2.1 of NIST
 597 Special Publication 800-56Ar2 and shall be the concatenation:

598 *AlgorithmID* || value of originator-system-title || length of transaction-id || value of
 599 transaction-id || value of recipient-system-title

600 where:

- 601 • *AlgorithmID* is that for AES-GCM-128 and so has a value 0x60857406080300, as
 602 specified by section 9.2.3.4.6.5 of the Green Book; and
- 603 • length of transaction-id has the value 0x09.

604 **4.3.3.4 Calculating the Initialization Vector for GCM and GMAC**

605 In relation to Remote Party Messages, Smart Metering Entities shall use a 96 bit Initialization
 606 Vector (IV) for the GCM and GMAC algorithms as defined in *NIST Special Publication 800-*
 607 *38D*. The IV shall be the concatenation

608 *FixedField* || *InvocationField*

609 where:

- 610 • *FixedField* = the Entity Identifier of the Smart Metering Entity that is creating, or has
 611 created, the Cryptographic Protection; and
- 612 • *InvocationField* = 0x00000000.

613 The DLMS COSEM Authentication Key (AK), as defined in the Green Book, shall not be
 614 present.

615 **4.3.3.4.1 Other input parameters to MAC and Encryption / Decryption operations -
 616 informative**

617 Other input parameters for MAC, Encryption and Decryption are not specified in this Section
 618 4.3.3 because they vary dependent on a number of factors. These other input parameters
 619 are listed in tables of the same format as Table 4.3.3.4.1 and their values are specified in
 620 each part of the GBCS where such an operation is specified.

621 The template for such tables is the Table 4.3.3.4.1. Please note that this table does not
 622 contain any values as it is a template only.

Input Parameter	Value	Note
To calculate the Shared Secret ('Z') input to the KDF:		
Private Key Agreement Key		
Public Key Agreement Key		
The other input to the KDF ('OtherInfo') shall be calculated according to the requirements of Section 4.3.3.3.		
As input to the GMAC function, the IV shall be constructed according to the requirements of Section 4.3.3.4, the Plaintext shall be empty and:		
Additional Authenticated Data shall be the		

Input Parameter	Value	Note
concatenation:		

623 Table 4.3.3.4.1: Template for other input parameters

624 **4.3.3.4.2 Size of MAC**

625 The bit length of the MAC shall be 96.

5 Remote Party Message Construction, Protection and Verification – informative

Much of the content, processing and structure of Remote Party Messages is common across multiple Messages. The GBCS lays out such common requirements. This is to allow Use Cases to detail only those requirements that are specific to the Message(s) covered by that Use Case.

5.1 Common Message Structures - informative

Parts of the structure and content of Remote Party Messages are common across multiple Remote Party Messages. These common parts of the structure and content are laid out in Section 7 of this GBCS. Section 7 also lays out specific requirements for DLMS COSEM and ZSE compliance for Devices compliant with this GBCS.

Note that Remote Party Messages in this GBCS are all constructed using aggregation structures. The GBCS does not allow for more granular message structures (e.g. for DLMS COSEM, individual set, get or action messages).

5.2 Common Encryption and Decryption approach - informative

The content and processing of fields in relation to Confidentiality shall be common across all parts of Messages requiring such protections. Where specified in a Use Case, a Remote Party Message may contain one or more encrypted parts. For such requirements, the corresponding Authenticated Encryption and Authenticated Decryption shall always be undertaken using the approach laid out in Section 8.

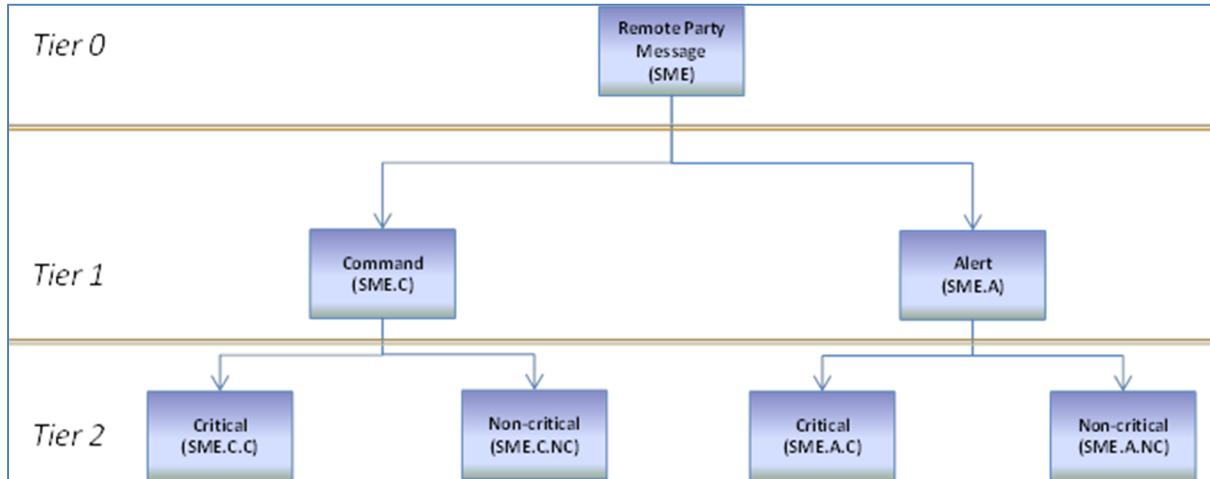
Note that the GBCS does not require Encryption of the whole of a Message.

5.3 Message Categories - informative

The content and processing of fields related to integrity, authenticity and non-repudiation varies according to whether:

- the Message is a Command, Response or Alert; and
- the Message is a Critical Message or not.

This leads to groupings which are referred to as Message Categories. Message Categories are structured in a hierarchical way, with the more generally applicable categories being at the tiers of the hierarchy with lower numbers. A category which is derived from another category (i.e. in a tier with a higher number) is called a subordinate Message Category. A category from which another category is derived (i.e. in a tier with a lower number) is called a superordinate Message Category. Figure 5.3 summarises the hierarchy.



660
 661 Figure 5.3: Message Categories Note that the 'Command' part of the hierarchy covers
 662 requirements for both the Command and the corresponding
 663 Response. Except in certain error cases (e.g. cryptographic processing failure), a Command
 664 always leads to a Response.

665 Section 6 is structured according to the hierarchy at Figure 5.3.

666 5.4 Common Message Processing steps - informative

667 A common set of stages for Remote Party Message processing is used in this GBCS and
 668 the Use Cases, except for Variant Messages¹⁵. Variant Messages include Security
 669 Credentials and Prepayment Top Up related Messages.

670 The common set of stages for Commands is shown in Table 5.4a.

Name of Stage	Summary of the stage	Responsible Smart Metering Entity
i. Command Construction	The Command is fully populated, apart from cryptographic fields	N/A The entity undertaking this phase is not known to the Device Although not apparent to the Device, the DSP's Transform Service would normally undertake such construction for DCC managed Devices
ii. Command Cryptographic Protection I	This stage is only needed where a Remote Party, other than the Access Control Broker, is required to add Cryptographic Protection to the Command. So for digital signing of Critical Commands only	Known Remote Party
iii. Command Cryptographic Protection II	The Access Control Broker adds its Cryptographic Protection to the Message. This is by way of the ACB adding a MAC	Access Control Broker
iv. Command Authentication and Integrity Verification	The Device undertakes the range of checks needed, including those to ensure authenticity of the sender and integrity of the Message. This includes checking the Identifiers and Counter in the Command and	Device

¹⁵ See Mapping Table for identification of Variant Messages

Name of Stage	Summary of the stage	Responsible Smart Metering Entity
	verifying the Access Control Broker's MAC	

671 Table 5.4a: Common stages for Commands That common set of stages for Responses is
 672 shown in Table 5.4b.

Name of Stage	Summary of the stage	Responsible Smart Metering Entity
v. Response Construction	The Response is fully populated by the Device, apart from cryptographic fields	Device
vi. Response Cryptographic Protection	The Device adds the required Cryptographic Protection to the Response	Device
vii. Response Recipient Verification	The Remote Party (Parties) can undertake the range of checks, including those to ensure authenticity of the sender and integrity of the Message	Remote Party named in the Response

673 Table 5.4b: Common stages for Responses
 674 That common set of stages for Alerts is shown in Table 5.4c.

Name of Stage	Summary of the stage	Responsible Smart Metering Entity
viii. Alert Construction	The Alert is fully populated by the Device, apart from cryptographic fields	Device
ix. Alert Cryptographic Protection	The Device adds the required cryptographic fields to the Alert	Device
x. Alert Recipient Verification	The Remote Party (Parties) can undertake the range of checks, including those to ensure the authenticity of the sender and integrity of the Message	Remote Party named in the Alert

675 Table 5.4c: Common stages for Alerts The generic processing applied to Commands and their
 676 Responses (in relation to integrity, authenticity and non-repudiation) in a Message Category
 677 is summarised in Table 5.4d.

	Command Construction	Command Cryptographic Protection I	Command Cryptographic Protection II	Command Authenticity and Integrity Verification	Response Construction	Response Cryptographic Protection	Response Recipient Cryptographic Verification
Responsible Party	Not known to Device	Known Remote Party	Access Control Broker	Smart Metering Device	Smart Metering Device	Smart Metering Device	Remote Party as named in the response
1 – Command (SME.C)	Commands contain sender ID, recipient ID and a Counter	-	Applies a MAC for the Device	Device checks Identifiers, checks the Counter and validates the MAC	Responses contain sender ID, recipient ID and a Counter	-	Can check Identifiers and the Counter
2 – Critical (SME.C.C)		Digitally signed				PLUS: Digitally signed	PLUS: Can verify digital signature
1 – Command (SME.C)	Commands contain sender ID, recipient ID and a Counter	-	Applies a MAC for the Device	Device checks Identifiers, checks the Counter and validates the MAC	Responses contain sender ID, recipient ID and a Counter	-	Can check Identifiers and the Counter
2 – Non-critical (SME.C.NC)						PLUS: Applies a MAC for the KRP	PLUS: Can verify the MAC

678 The generic processing applied to Alerts in a Message Category is summarised in Table
 679 5.4e.

	Alert Construction	Alert Cryptographic Protection	Alert Recipient Cryptographic Verification
Responsible Party	Smart Metering Device	Smart Metering Device	Remote Party as named in the response
1 – Alert (SME.A)	Alerts contain sender ID, recipient ID and a Counter	-	Can check Identifiers and the Counter
2 – Critical (SME.A.C)		Digitally signed	PLUS: Can verify digital signature
1 – Alert (SME.A)	Alerts contain sender ID, recipient ID and a Counter	-	Can check Identifiers and the Counter
2 – Non-critical (SME.A.NC)		Applies a MAC for the KRP	PLUS: Can verify the MAC

680
 681 Table 5.4e: Generic Alert processing

5.5 Common processing stages and requirements for Devices operated through the DCC - informative

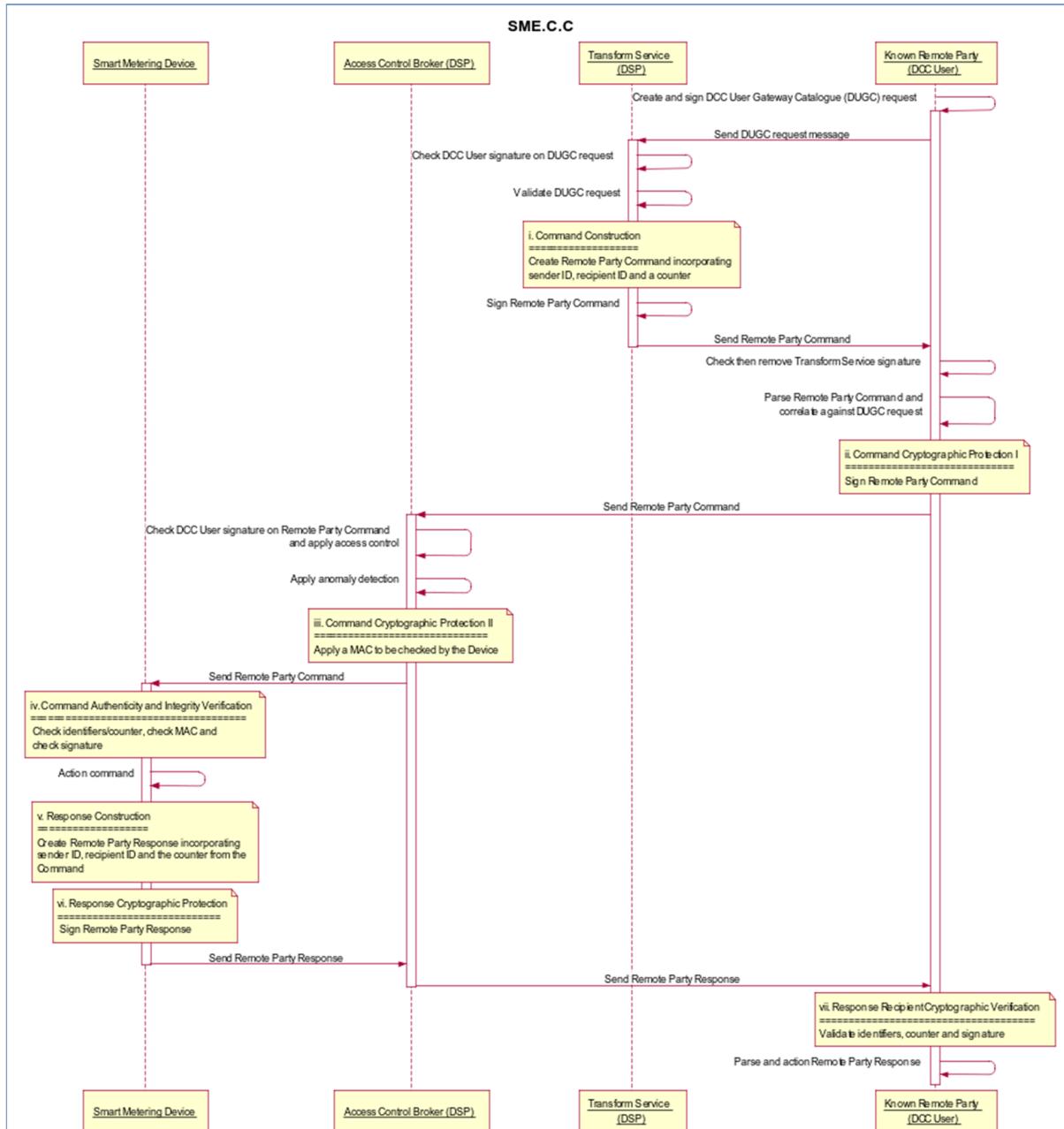
684 The sequence diagrams in the figures in this Section 5.5 illustrate the generic processing
 685 stages and common processing requirements, where a Device is operated via the DCC, for
 686 each of:

- 687 • SME.C.C: Critical Remote Party Command to a Device and the corresponding Remote
 688 Party Response (Figure 5.5a);
- 689 • SME.C.NC: non Critical Remote Party Command to a Device from a Known Remote
 690 Party and the corresponding Remote Party Response (Figure 5.5b);
- 691 • SME.A.C: Critical Alert from a Device (Figure 5.5c); and

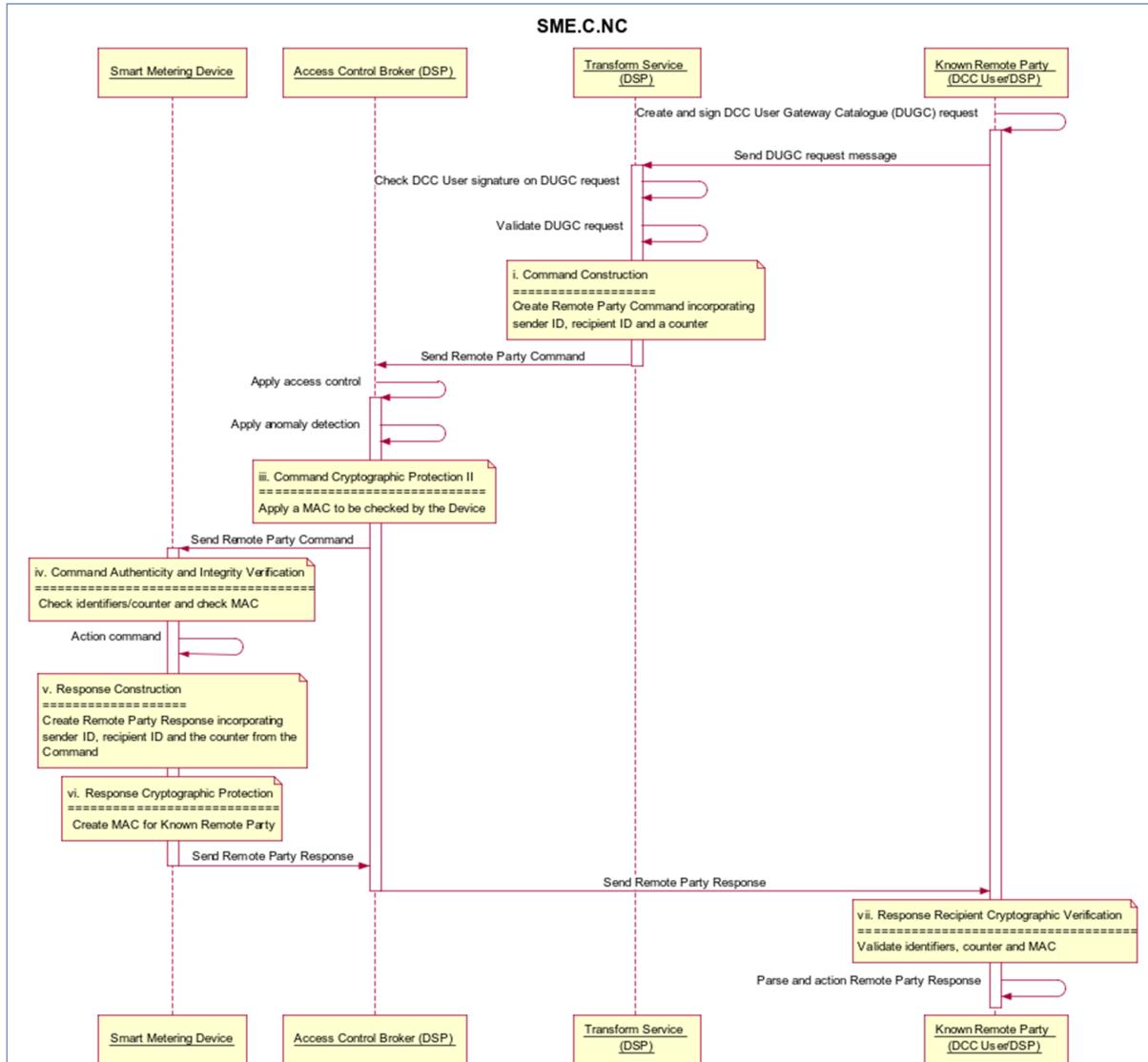
- 692 • SME.A.NC: non Critical Alert from a Device (Figure 5.5d).

693 Note that only those parts of the sequence diagrams within yellow notes boxes are within the
 694 scope of the GBCS. The steps outside such boxes are provided for context and, where
 695 mandated, are mandated through mechanisms outside the GBCS, for example the Smart
 696 Energy Code.

697 For DCC managed Devices, the DSP would operate the services that provide (1) Access
 698 Control Broker, (2) Transform Service and (3) Transitional Change of Supplier. The CSPs
 699 would fulfil the role of WAN Provider.

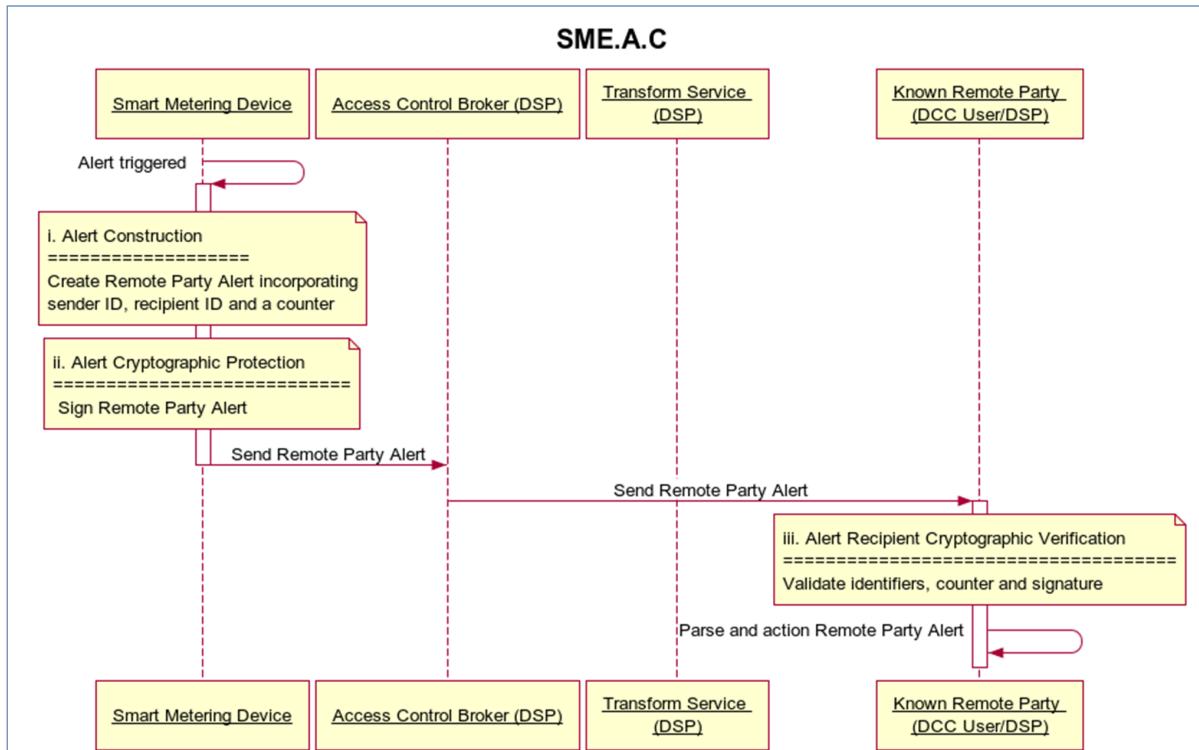


700 Figure 5.5a: Sequence diagram for processing Critical Remote Party Commands and Responses



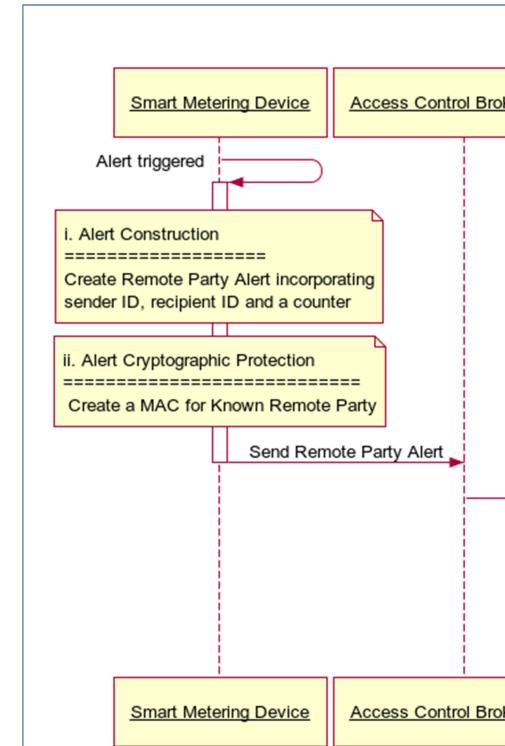
701

Figure 5.5b: Sequence diagram for processing non Critical Remote Party Commands and Responses



702

Figure 5.5c: Sequence diagram for processing Critical Remote Party Alerts



703

Figure 5.5d: Sequence diagram for processing non Critical Remote Party Alerts

6 Message Categories

Requirements for the content and processing of fields in Remote Party Messages:

- related to integrity, Authenticity and non-repudiation; and
- common across groups of Remote Party Messages.

are laid out in this Section 6. Such groupings of Remote Party Messages are referred to as Message Categories.

Commands sent by a PPMID to a GSME and Responses to such Commands have requirements similar to Message Categories, and common requirements for this group of Messages are also laid out in this Section 6.

6.1 Introduction - informative

Please see the Mapping Table for the mapping of Use Cases to the Message Categories in this Section 6.

6.2 Message Category SME.C

6.2.1 Definitions

The superordinate Message Category for SME.C is SME.

For a Message to be of Message Category SME.C it shall be a Command to a Device which is a Remote Party Message, or a Command from a PPMID to a GSME, or a Response to such Commands.

All SME.C Commands and any corresponding Response shall comply with the requirements of this Section 6.2 which covers:

- generation of a MAC by the Access Control Broker / PPMID and verification of that MAC by the Device; and
- validation by the Device of the Message Identifier.

6.2.2 Processing Stages

The processing of each SME.C Command shall have the stages set out in Table 6.2.2a.

Stage	Responsible Smart Metering Entity
xi. Command Construction	The entity undertaking this phase is not known to the Device
xii. Command Cryptographic Protection I	Known Remote Party
xiii. Command Cryptographic Protection II	Access Control Broker / PPMID
xiv. Command Authenticity and Integrity Verification	Device

Table 6.2.2a: SME.C Command Processing Stages For a Command, should any of the checks required in the Command Authenticity and Integrity Verification step fail, the Device shall take the steps laid out in Section 6.2.4.2. Otherwise the stages of processing set out in Table 6.2.2b shall be undertaken.

Stage	Responsible Smart Metering Entity
xv. Response Construction	Device
xvi. Response Cryptographic Protection	Device

xvii. Response Recipient Verification	Remote Party named in the Response, or the PPMID named in the Response
---------------------------------------	--

Table 6.2.2b: SME.C Response Processing Stages

6.2.2.1 Processing stages defined in the superordinate Message Category

There are no processing stages defined in the superordinate Message Category (SME).

6.2.2.2 Processing stages defined in subordinate Message Categories

There are no requirements for the following processing stages as they are wholly defined in subordinate Message Categories:

- Command Construction;
- Command Cryptographic Protection I;
- Response Construction;
- Response Cryptographic Protection; and
- Response Recipient Verification.

6.2.3 Command Cryptographic Protection II

Requirements in this Section 6.2.3 for Command Cryptographic Protection II shall apply to Message Category SME.C and all subordinate categories.

For Remote Party Commands, the Access Control Broker shall calculate the Access Control Broker to Device Message Authentication Code (ACB-SMD MAC) using the parameters in Table 6.2.3a.

Input Parameter	Value	Note
To calculate the Shared Secret ('Z') input to the KDF:		
Private Key Agreement Key	Access Control Broker's	
Public Key Agreement Key	Device's	As identified by the Business Target ID in Message Identifier
The other input to the KDF ('OtherInfo') shall be calculated according to the requirements of Section 4.3.3.3.		
As input to the GMAC function, the IV shall be constructed according to the requirements of Section 4.3.3.4, the Plaintext shall be empty and:		
Additional Authenticated Data shall be the concatenation:	Where a KRP Signature is present: 0x11 Grouping Header Command Payload 0x40 KRP Signature Where a KRP Signature is not present: 0x11 Grouping Header Command Payload 0x00	

Table 6.2.3a: Calculation of Access Control Broker to Device MAC The ACB-SMD MAC for incorporation in the Command shall only be calculated once all fields of the Command are populated, as per requirements for the Command Construction and Command Cryptographic Protection I stages for the Message in question.

753 For HAN Only Commands from the PPMID to a GSME, the PPMID shall calculate the
 754 PPMID to GSME Message Authentication Code (PPMID-GSME MAC) using the parameters
 755 in Table 6.2.3b.

Input Parameter	Value	Note
To calculate the Shared Secret ('Z') input to the KDF:		
Private Key Agreement Key	PPMID's	
Public Key Agreement Key	GSME's	As held by the PPMID in the GSME Trust Anchor Cell
The other input to the KDF ('OtherInfo') shall be calculated according to the requirements of Section 4.3.3.3.		
As input to the GMAC function, the IV shall be constructed according to the requirements of Section 4.3.3.4, the Plaintext shall be empty and:		
Additional Authenticated Data shall be the concatenation:	0x11 Grouping Header Command Payload 0x00	

756 Table 6.2.3b: Calculation of PPMID-GSME MAC The PPMID-GSME MAC for incorporation in
 757 the Command shall only be calculated once all fields of the Command are populated, as per
 758 requirements for the Command Construction and Command Cryptographic Protection I
 759 stages for the Message in question.

6.2.4 Command Authenticity and Integrity Verification

760 Requirements in this Section 6.2.4 shall apply to Message Category SME.C and all
 761 subordinate categories.

6.2.4.1 Checks to be undertaken

762 The Device shall undertake the checks in the sequence set out in this Section 6.2.4.1 before
 763 undertaking any other processing of the Command.

6.2.4.1.1 Message Identifier Validation

764 The Device shall verify that:

- 765 5. the Business Target ID in the Command has the same value as the Device's Entity Identifier;
- 766 6. the Message Code is for a Message that the Device is capable of processing, according to the associated Use Case;
- 767 7. the contents of the Message conform to the message formatting and structure requirements of this GBCS and the associated Use Case; and
- 768 8. the Business Originator ID in the Command has the same value as the Entity Identifier held by the Device within a Trust Anchor Cell, where the Smart Metering Entity associated with that Trust Anchor Cell is allowed to request execution of a Command of this type, as specified by the Message Code in the Command and the Mapping Table ('Use Case reference' worksheet Message Code columns).

6.2.4.1.2 ACB-SMD MAC Verification

769 To verify the ACB-SMD MAC in Remote Party Commands, the Device shall calculate a MAC
 770 using the parameters in Table 6.2.4.1.2 and ensure the MAC so calculated has the same
 771 value as the ACB-SMD MAC.

Input Parameter	Value	Note
-----------------	-------	------

Input Parameter	Value	Note
To calculate the Shared Secret ('Z') input to the KDF:		
Private Key Agreement Key	Device's	
Public Key Agreement Key	Access Control Broker's	As held by the Device in the Trust Anchor Cell {accessControlBroker, keyAgreement, management}
The other input to the KDF ('OtherInfo') shall be calculated according to the requirements of Section 4.3.3.3.		
As input to the GMAC function, the IV shall be constructed according to the requirements of Section 4.3.3.4, the Plaintext shall be empty and:		
Additional Authenticated Data shall be the concatenation:	Where a KRP Signature is present: 0x11 Grouping Header Command Payload 0x40 KRP Signature Where a KRP Signature is not present: 0x11 Grouping Header Command Payload 0x00	

Table 6.2.4.1.2: MAC calculation for ACB-SMD MAC verification

6.2.4.1.3 PPMID-GSME MAC Verification

To verify the PPMID-GSME MAC in HAN Only Commands from a PPMID to a GSME, the Device shall calculate a MAC using the parameters in Table 6.2.4.1.3 and ensure the MAC so calculated has the same value as the PPMID-GSME MAC.

Input Parameter	Value	Note
To calculate the Shared Secret ('Z') input to the KDF:		
Private Key Agreement Key	GSME's	
Public Key Agreement Key	PPMID's	As held by the GSME in the PPMID Trust Anchor Cell.
The other input to the KDF ('OtherInfo') shall be calculated according to the requirements of Section 4.3.3.3.		
As input to the GMAC function, the IV shall be constructed according to the requirements of Section 4.3.3.4, the Plaintext shall be empty and:		
Additional Authenticated Data shall be the concatenation:	0x11 Grouping Header Command Payload 0x00	

Table 6.2.4.1.3: MAC calculation for PPMID-GSME MAC verification

6.2.4.2 Processing based on the outcome of checks

If the Message requires 'Protection Against Replay' according to the corresponding Use Case, the Device shall ensure that the Originator Counter in the Command has a value that is greater than the value held by the Device for this type of Command in the corresponding Execution Counter.

Where this check or any of the other prior required checks for this type of Command have failed, the Device shall:

- 794 • generate an entry in the Security Log recording failed Authentication;
- 795 • discard the Command without execution and without sending a Response; and
- 796 • send an Alert notifying the failed Authentication, constructed as specified in Section 6.7,
- 797 populated with the relevant Alert Code from Section 16 (the one of 0x801E, 0x8030 or
- 798 0x803D that is required), to the Known Remote Party specified in Section 16. If the
- 799 Device is an ESME or a CHF, the Alert Payload shall be a DLMS COSEM Alert Payload.
- 800 Otherwise, the Alert Payload shall be a GBZ Alert Payload.

801 Where all of the checks required to be undertaken by the Device have succeeded, the
802 Device shall:

- 803 • if the Message requires 'Protection Against Replay' according to the corresponding Use
804 Case, update the Execution Counter for a Command with the Message Code contained
805 within the Message from the Remote Party Role identified by the Message, to the value
806 of the Originator Counter in the Command; and
- 807 • where the Command contains one or more activation times, set the corresponding
808 activation times stored on the Device to the relevant values detailed in Section 9.2.2.4,
809 and process the Command and produce a Response..

6.3 Message Category SME.C.C

6.3.1 Definitions

812 The superordinate Message Category for SME.C.C is SME.C.

813 For a Message to be of Message Category SME.C.C it shall be:

- 814 • a subordinate Message Category of Message Category SME.C;
- 815 • from or to a Remote Party; and
- 816 • a Critical Message.

817 A Device shall only be capable of processing the Critical Commands laid out in the GBCS.

818 All SME.C.C Commands and any corresponding Response shall comply both with the
819 requirements for SME.C Messages and with the requirements of this Section 6.3 which
820 covers:

- 821 • Digital Signing of the Command by the Known Remote Party;
- 822 • verification of the Digital Signature in the Command by the Device;
- 823 • Digital Signing of the Response by the Device; and
- 824 • verification of the Digital Signature in the Response by the Known Remote Party.

6.3.2 Processing stages

6.3.2.1 Processing stages defined in the superordinate Message Category

827 There are no requirements additional to those of the superordinate Message Category
828 (SME.C) for the Command Cryptographic Protection II stage.

6.3.2.2 Processing stages defined in subordinate categories

830 There are no requirements for the following processing stages as they are wholly defined in
831 subordinate categories:

- 832 • Command Construction; and
- 833 • Response Construction.

6.3.3 Command Cryptographic Protection I

Requirements in this Section 6.3.3 shall apply to Message Category SME.C.C and all subordinate categories.

The Remote Party originating the Command shall generate a Known Remote Party Signature (KRP Signature) for the Command.

The KRP Signature, for incorporation in the Command, shall only be generated once all fields of the Command Payload and Grouping Header are populated as per the requirements for the Command Construction stage, for the Message in question.

The KRP Signature shall be calculated across those fields of Grouping Header specified in Section 7.2.7 and all fields of the Command Payload, as specified in Section 7.2.7.

The Remote Party shall use its Private Digital Signing Key to generate the KRP Signature.

6.3.4 Command Authenticity and Integrity Verification

Requirements in this Section 6.3.4 shall apply to Message Category SME.C.C and all subordinate categories.

The Device shall undertake the checks set out in this Section 6.3.4:

- only after all checks in Section 6.2.4.1 have been successfully completed; and
- before undertaking any other processing of the Command.

The Device shall use the Command Payload, Grouping Header and the Public Digital Signing Key of the Remote Party identified by the checks in Section 4.3.2.7.2 for Digital Signature verification of the KRP Signature.

The actions laid out in Section 6.2.4.2 shall then apply, as required by the success or failure of the Digital Signature verification.

6.3.5 Response Cryptographic Protection

The Device creating the Response shall generate a Device Signature (SMD Signature) for the Response.

The SMD Signature, for incorporation in the Response, shall only be generated once all fields of the Response Payload and Grouping Header are populated, as per requirements for the Response Construction stage, for the Message in question.

The SMD Signature shall be calculated across those fields of Grouping Header specified in Section 7.2.7 and all fields of the Response Payload, as specified in Section 7.2.7.

The Device shall use its Private Digital Signing Key to generate the SMD Signature.

6.3.6 Response Recipient Verification

A Remote Party may verify the SMD Signature in the Response by using the Response Body and the Public Digital Signing Key for the Device identified in the Response.

6.4 Message Category SME.C.NC

6.4.1 Definitions

For a Message to be of Message Category SME.C.NC, it shall be:

- a subordinate Message Category of Message Category SME.C;
- from or to a Remote Party; and
- not a Critical Message.

874 All SME.C.NC Commands and any corresponding Response shall comply both with the
 875 requirements for SME.C Messages and with the requirements of this Section 6.4 which
 876 covers:

- 877 • generation by the Device of a MAC for the Response; and
- 878 • verification of that MAC by the intended recipient of the Response.

879 **6.4.2 Processing stages**

880 **6.4.2.1 Processing stages defined in the superordinate Message Category**

881 There are no requirements additional to those of the superordinate Message Category
 882 (SME.C) for the Command Cryptographic Protection II processing stage.

883 **6.4.2.2 Processing stages defined in subordinate categories**

884 There are no requirements for the following processing stages as they are wholly defined in
 885 subordinate categories:

- 886 • Command Construction; and
- 887 • Response Construction.

888 **6.4.3 Command Cryptographic Protection I**

889 There are no additional requirements at the Command Cryptographic Protection I stage
 890 applicable to all Messages of Message Category SME.C.NC and any subordinate Message
 891 Category.

892 **6.4.4 Command Authenticity and Integrity Verification**

893 There are no additional requirements at the Command Authenticity and Integrity Verification
 894 stage applicable to all Messages of Message Category SME.C.NC and any subordinate
 895 Message Category.

896 **6.4.5 Response Cryptographic Protection**

897 Requirements in this Section 6.4.5 shall apply to Message Category SME.C.NC and all
 898 subordinate categories.

899 The Device shall calculate the Device to Known Remote Party MAC (SMD-KRP MAC) using
 900 the parameters in Table 6.4.5.

Input Parameter	Value	Note
To calculate the Shared Secret ('Z') input to the KDF:		
Private Key Agreement Key	Device's	
Public Key Agreement Key	Known Remote Party's	As held by the Device in the relevant Trust Anchor Cell {remotePartyRole, keyAgreement, management}. The relevant Cell will contain Business Originator ID as specified in Message Identifier.
The other input to the KDF ('OtherInfo') shall be calculated according to the requirements of Section 4.3.3.3.		
As input to the GMAC function, the IV shall be constructed according to the requirements of Section 4.3.3.4, the Plaintext shall be empty and:		
Additional Authenticated Data	0x11 Grouping Header	

Input Parameter	Value	Note
shall be the concatenation:	Response Payload 0x00	

901 Table 6.4.5: Calculation of Device to Known Remote Party MAC The SMD-KRP MAC for
 902 incorporation in the Response shall only be calculated once all fields of the Response,
 903 except for the SMD-KRP MAC itself, are populated as per requirements for the Response
 904 Construction stage, for the Message in question.

6.4.6 Response Recipient Verification

905 Requirements in this Section 6.4.6 shall apply to Message Category SME.C.NC and all
 906 subordinate categories.

908 The Remote Party, as identified by the Business Originator ID in the Response, may validate
 909 the SMD-KRP MAC in the Response by calculating a MAC using the parameters in Table
 910 6.4.6 and comparing the MAC to the SMD-KRP MAC.

Input Parameter	Value	Note
To calculate the Shared Secret ('Z') input to the KDF:		
Private Key Agreement Key	Known Remote Party's	
Public Key Agreement Key	Device's	As identified by the Business Target ID in Message Identifier
The other input to the KDF ('OtherInfo') shall be calculated according to the requirements of Section 4.3.3.3.		
As input to the GMAC function, the IV shall be constructed according to the requirements of Section 4.3.3.4, the Plaintext shall be empty and:		
Additional Authenticated Data shall be the concatenation:	0x11 Grouping Header Response Payload 0x00	

911 Table 6.4.6: MAC calculation for SMD-KRP MAC validation

6.5 Message Category SME.C.PPMID-GSME

6.5.1 Definitions

914 For a Message to be of Message Category SME.C.PPMID-GSME, it shall be:

- a subordinate Message Category of Message Category SME.C; and
- a Message between a PPMID and a GSME.

917 All SME.C.PPMID-GSME Commands and any corresponding Response shall comply both
 918 with the requirements for SME.C Messages and with the requirements of this Section 6.4
 919 which covers:

- generation by the Device of a MAC for the Response; and
- verification of that MAC by the intended recipient of the Response.

6.5.2 Processing stages

6.5.2.1 Processing stages defined in the superordinate Message Category

924 There are no requirements additional to those of the superordinate Message Category
 925 (SME.C) for the Command Cryptographic Protection II processing stage.

926 **6.5.2.2 Processing stages defined in subordinate categories**

927 There are no requirements for the following processing stages as they are wholly defined in
 928 subordinate categories:

- 929 • Command Construction; and
- 930 • Response Construction.

931 **6.5.3 Command Cryptographic Protection I**

932 There are no additional requirements at the Command Cryptographic Protection I stage
 933 applicable to all Messages of Message Category SME.C.PPMID-GSME and any subordinate
 934 Message Category.

935 **6.5.4 Command Authenticity and Integrity Verification**

936 There are no additional requirements at the Command Authenticity and Integrity Verification
 937 stage applicable to all Messages of Message Category SME.C.PPMID-GSME and any
 938 subordinate Message Category.

939 **6.5.5 Response Cryptographic Protection**

940 Requirements in this Section 6.5.5 shall apply to Message Category SME.C.PPMID-GSME
 941 and all subordinate categories.

942 The GSME shall calculate the GSME to PPMID MAC (GSME-PPMID MAC) using the
 943 parameters in Table 6.5.5.

Input Parameter	Value	Note
To calculate the Shared Secret ('Z') input to the KDF:		
Private Key Agreement Key	Device's	
Public Key Agreement Key	PPMID's	As held by the GSME in the PPMID Trust Anchor Cell
The other input to the KDF ('OtherInfo') shall be calculated according to the requirements of Section 4.3.3.3.		
As input to the GMAC function, the IV shall be constructed according to the requirements of Section 4.3.3.4, the Plaintext shall be empty and:		
Additional Authenticated Data shall be the concatenation:	0x11 Grouping Header Response Payload 0x00	

944 Table 6.5.5: Calculation of GSME-PPMID MAC

945 The GSME-PPMID MAC for incorporation in the Response shall only be calculated once all
 946 fields of the Response, except for the GSME-PPMID MAC itself, are populated as per
 947 requirements for the Response Construction stage, for the Message in question.

948 **6.5.6 Response Recipient Verification**

949 Requirements in this Section 6.5.6 shall apply to Message Category SME.C.PPMID-GSME
 950 and all subordinate categories.

951 The PPMID, as identified by the Business Originator ID in the Response, shall validate the
 952 GSME-PPMID MAC in the Response by calculating a MAC using the parameters in Table
 953 6.5.6 and comparing the MAC to the GSME-PPMID MAC.

Input Parameter	Value	Note
To calculate the Shared Secret ('Z') input to the KDF:		

Input Parameter	Value	Note
Private Key Agreement Key	PPMID's	
Public Key Agreement Key	GSME's	As held by the PPMID in the GSME Trust Anchor Cell
The other input to the KDF ('OtherInfo') shall be calculated according to the requirements of Section 4.3.3.3.		
As input to the GMAC function, the IV shall be constructed according to the requirements of Section 4.3.3.4, the Plaintext shall be empty and:		
Additional Authenticated Data shall be the concatenation :	0x11 Grouping Header Response Payload 0x00	

954 Table 6.5.6: MAC calculation for GSME-PPMID MAC validation

955 6.6 Message Category SME.A

956 6.6.1 Definitions

957 The superordinate Message Category for SME.A is SME.

958 For a Message to be of Message Category SME.A it shall be an Alert from a Device which is
959 addressed to a Remote Party.

960 There are no common requirements that shall be applied to all Messages of Message
961 Category SME.A.

962 6.6.2 Processing Stages

963 The processing of each SME.A Alert shall have the stages set out in Table 6.6.2:

Stage	Responsible Smart Metering Entity
xviii. Alert Construction	Device
xix. Alert Cryptographic Protection	Device
xx. Alert Recipient Verification	Remote Party named in the Alert.

964 Table 6.6.2: SME.A Processing Stages

965 6.6.2.1 Processing stages defined in the superordinate Message Category

966 There are no processing stages defined in the superordinate Message Category (SME).

967 6.6.2.2 Processing stages defined in subordinate categories

968 There are no requirements for the following processing stages as they are wholly defined in
969 subordinate categories:

- 970 • Alert Construction;
- 971 • Alert Cryptographic Protection; and
- 972 • Alert Recipient Verification.

973 6.7 Message Category SME.A.C

974 6.7.1 Definitions

975 For a message to be categorised as Message Category SME.A.C, it shall be:

- 976 • a subordinate Message Category of Message Category SME.A; and
- 977 • a Critical Message.

978 All SME.A.C Messages shall comply both with the requirements for SME.A Messages and
979 with the requirements of this Section 6.7 which covers:

- 980 • Digital Signing of the Alert by the Device; and
981 • Verification of the Digital Signature in the Alert by the Remote Party.

982 **6.7.2 Processing stages**

983 **6.7.2.1 Processing stages defined in the superordinate Message Category**

984 There are no processing stages defined in the superordinate Message Category (SME.A).

985 **6.7.2.2 Processing stages defined in subordinate categories**

986 There are no requirements for the Alert Construction processing stage as they are wholly
987 defined in subordinate categories.

988 **6.7.3 Alert Cryptographic Protection**

989 Requirements in this Section 6.7.3 shall apply to Message Category SME.A.C and all
990 subordinate categories.

991 The Device creating the Alert shall generate a Device Signature (SMD Signature) for the
992 Alert.

993 The SMD Signature, for incorporation in the Alert, shall only be generated once all fields of
994 the Alert Payload and Grouping Header are populated, as per requirements for the Alert
995 Construction stage for the Message in question.

996 The SMD Signature shall be calculated across those fields of Grouping Header and all fields
997 of the Alert Payload, both as specified in Section 7.2.7.

998 The Device shall use its Private Digital Signing Key to generate the SMD Signature.

999 **6.7.4 Alert Recipient Verification**

1000 Requirements in this Section 6.7.4 shall apply to Message Category SME.A.C and all
1001 subordinate categories.

1002 A Remote Party may verify the SMD Signature in the Alert by using the Alert Payload,
1003 Grouping Header and the Public Digital Signing Key for the Device, as identified in the Alert.

1004 **6.8 Message Category SME.A.NC**

1005 **6.8.1 Definitions**

1006 For a Message to be of Message Category SME.A.NC it shall be:

- 1007 • a subordinate Message Category of Message Category SME.A; and
1008 • not a Critical Message.

1009 All SME.A.NC Messages shall comply both with the requirements for SME.A Messages and
1010 with the requirements of this Section 6.8 which covers:

- 1011 • generation by the Device of a MAC for the Alert and validation of that MAC by the
1012 intended recipient of the Alert.

1013 **6.8.2 Processing stages**

1014 **6.8.2.1 Processing stages defined in the superordinate Message Category**

1015 There are no processing stages defined in the superordinate Message Category (SME.A).

1016 **6.8.2.2 Processing stages defined in subordinate categories**

1017 There are no requirements for the Alert Construction processing stage as they are wholly
1018 defined in subordinate categories.

1019 **6.8.3 Alert Cryptographic Protection**

1020 Requirements in this Section 6.8.3 shall apply to Message Category SME.A.NC and all
1021 subordinate categories.

1022 The Device shall calculate the Device to Known Remote Party MAC (SMD-KRP MAC) using
1023 the parameters in Table 6.8.3.

Input Parameter	Value	Note
To calculate the Shared Secret ('Z') input to the KDF:		
Private Key Agreement Key	Device's	
Public Key Agreement Key	Known Remote Party's	As held by the Device in the relevant Trust Anchor Cell {remotePartyRole, keyAgreement, management}. The relevant Trust Anchor Cell will contain Business Originator ID as specified in Message Identifier.
The other input to the KDF ('OtherInfo') shall be calculated according to the requirements of Section 4.3.3.3.		
As input to the GMAC function, the IV shall be constructed according to the requirements of Section 4.3.3.4, the Plaintext shall be empty and:		
Additional Authenticated Data shall be the concatenation:	0x11 Grouping Header Alert Payload 0x00	

1024 Table 6.8.3: Calculation of the Device to Known Remote Party MAC

1025 The SMD-KRP MAC for incorporation in the Alert shall only be calculated once all fields of
1026 the Alert, except for the SMD-KRP MAC itself, are populated as per requirements for the
1027 Alert Construction stage, for the Message in question.

1028 **6.8.4 Alert Recipient Verification**

1029 Requirements in this Section 6.8.4 shall apply to Message Category SME.A.NC and all
1030 subordinate categories.

1031 The Remote Party, as identified by the Business Originator ID in the Alert, may validate the
1032 SMD-KRP MAC in the Alert by calculating a MAC using the parameters in Table 6.8.4 and
1033 comparing the MAC to the SMD-KRP MAC.

Input Parameter	Value	Note
To calculate the Shared Secret ('Z') input to the KDF:		
Private Key Agreement Key	Known Remote Party's	
Public Key Agreement Key	Device's	As identified by the Business Originator ID in Message Identifier
The other input to the KDF ('OtherInfo') shall be calculated according to the requirements of Section 4.3.3.3.		
As input to the GMAC function, the IV shall be constructed according to the requirements of Section		

Input Parameter	Value	Note
4.3.3.4, the Plaintext shall be empty and:		
Additional Authenticated Data shall be the concatenation:	0x11 Grouping Header Alert Payload 0x00	

Table 6.8.4: MAC calculation for SMD-KRP MAC verification

7 Message structure and DLMS COSEM / ZSE / ASN.1 requirements

7.1 Introduction - informative

This Section 7:

- defines the structure of Remote Party Messages containing DLMS COSEM, ASN.1 and GBZ Payloads. A GBZ Payload is a Payload containing one or more ZSE messages;
- defines the structure of Messages between a PPMID and a GSME on the same SMHAN; and
- lays out specific requirements for DLMS COSEM and ZSE compliance to which Devices shall adhere.

Note that Remote Party Messages all use an aggregation structure which allows for multiple, protocol-specific instructions within the same Message. The aggregation structures are used for all Messages, are based on xDLMS access service, general signing service and general ciphering service formats, and provide protections across all types of Message payload (be they DLMS COSEM, ZSE or security related).

The GBCS does not provide more granular Message structures (e.g. for DLMS COSEM, individual set, get or action messages).

SMETS and CHTS require that the Critical Commands mandated by them (and so those defined in the GBCS) are the only Critical commands allowed. Devices may implement additional non Critical features only.

It should be noted that:

- SMETS only requires DLMS COSEM certification on the ESME;
- any action that the Known Remote Party takes to remedy a failure will need to factor in that some of the instructions succeeded and others did not;
- in ASN.1 notation, the signature field in the general-signing service is a variable length OCTET STRING. When encoded, this means that the length of the signature needs to be incorporated before the actual signature value. The length is either 64 (0x40) if a signature is present or 0 (0x00) if signature is not present;
- these requirements are to ensure that all Devices behave consistently and in the way required by originating Remote Party requests, including in error states; and
- the WAN Provider may read CHF Operational Data and CHF Configuration Data, with their CHTS meanings, using mechanisms other than those defined in this GBCS.

7.2 Remote Party Message Construction - general

This Section 7.2 shall apply to Messages which are of a Message Category that is not 'Variant'. For Messages of a Message Category that is 'Variant', this Section 7.2 shall only apply where explicitly stated in the Use Case, with the exception of Section 7.2.11 which shall apply to all Messages.

Except for elements detailed as being defined in the ZSE or ZCL specifications, the octet strings constructed in compliance with this Section 7 shall be in 'big endian' order according to IETF RFC 1700¹⁶. Elements detailed in this Section 7 as being defined in the ZSE or ZCL

¹⁶ <http://tools.ietf.org/html/rfc1700>

1074 specifications, shall be serialised into the corresponding parts of octet strings as defined in
 1075 the corresponding ZSE or ZCL specification.

1076 7.2.1 Commands

1077 Whether a Command requires a KRP Signature is specified in the corresponding Message
 1078 Category requirements in Section 6.

1079 Where a KRP Signature is required, a Remote Party Command received by a Device shall
 1080 be the concatenation:

1081 MAC Header || Grouping Header || Command Payload || 0x40 || KRP Signature || ACB-
 1082 SMD MAC

1083 Where a KRP Signature is not required, a Remote Party Command received by a Device
 1084 shall be the concatenation:

1085 MAC Header || Grouping Header || Command Payload || 0x00 || ACB-SMD MAC

1086 A HAN Only Command from a PPMID to a GSME shall be the concatenation:

1087 MAC Header || Grouping Header || Command Payload || 0x00 || PPMID-GSME MAC

1088 7.2.2 Responses

1089 Whether a Response requires an SMD Signature is specified in the corresponding Message
 1090 Category requirements in Section 6.

1091 Where a SMD Signature is required, a Remote Party Response shall be the concatenation:

1092 Grouping Header || Response Payload || 0x40 || SMD Signature

1093 Where a SMD Signature is not required, a Remote Party Response shall be the
 1094 concatenation:

1095 MAC Header || Grouping Header || Response Payload || 0x00 || SMD-KRP MAC

1096 A HAN Only Response from a GSME to a PPMID shall be the concatenation:

1097 MAC Header || Grouping Header || Response Payload || 0x00 || GSME-PPMID MAC

1098 7.2.3 Alerts

1099 Whether an Alert requires an SMD Signature is specified in the corresponding Message
 1100 Category requirements in Section 6.

1101 Where a SMD Signature is required, a Remote Party Alert shall be the concatenation:

1102 Grouping Header || Alert Payload || 0x40 || SMD Signature

1103 Where a SMD Signature is not required, a Remote Party Alert shall be the concatenation:

1104 MAC Header || Grouping Header || Alert Payload || 0x00 || SMD-KRP MAC

1105 7.2.4 Payload sequence and Break On Error

1106 All Message Payloads - Command Payloads, Response Payloads and Alert Payloads - shall:

- 1107 • only be constructed in the sequence specified in the corresponding Use Case;
- 1108 • only be processed in the sequence specified in the corresponding Use Case; and
- 1109 • be processed by a recipient Device on a Break On Error basis.

1110 Where a Command Payload contains multiple instructions, processing of further instructions
 1111 shall cease at the point any one instruction fails. In line with the DLMS COSEM Access
 1112 Services requirements, a Response shall contain one result for each instruction in the

1113 Command. The corresponding result in the Response Payload shall detail that instruction's
 1114 success or failure. The Response Payload shall explicitly detail a result for each of the
 1115 subsequent instructions that were not attempted. The results in the Response shall be in the
 1116 same order as the instructions in the Command.

1117 The specific result codes shall be as specified in the relevant ZSE / DLMS COSEM
 1118 document, or in this GBCS where standard-based error codes do not exist. Where
 1119 execution of instructions was not attempted due to the Break On Error requirement, the
 1120 response shall return:

- 1121 • for DLMS instructions, a Data-Access-Result / Action-Result of other-
 1122 reason;
 - 1123 • for ZCL / ZSE instructions, a ZCL / ZSE status value of FAILURE (0x01), with a
 1124 Command ID set to 0xFF.
- 1125 A ZSE command returning a status of 'NOT_FOUND' shall not be treated as a failure.

7.2.5 Message Construction – MAC Header

1126 The required components of the MAC Header shall be populated with the values as per
 1127 Table 7.2.5.¹⁷

MAC Header				
No	xDLMS Message Elements	Contents	Length (octets)	Note
	General-Ciphering	0xDD	1	xDLMS APDU tag for General-Ciphering (221 in decimal)
	transaction-id	0x00	1	A value for this element is not needed so the length field is 0x00
	originator-system-title	0x00	1	A value for this element is not needed so the length field is 0x00
	recipient-system-title	0x00	1	A value for this element is not needed so the length field is 0x00
	date-time	0x00	1	A value for this element is not needed so the length field is 0x00
	other-information	0x00	1	A value for this element is not needed so the length field is 0x00
	key-info	0x00	1	Key-info values are not present so encoded as 0x00
	ciphered-service			
	Length	Encoding(X)	Len(Encoding(X))	X shall be the length in octets of the subsequent parts of the Message after this Length value. This includes the security header, the DLMS APDU being protected and the MAC

¹⁷ See Green Book.

MAC Header				
No	xDLMS Message Elements	Contents	Length (octets)	Note
	security header			
	security control byte (SC)	0x11	1	<p>Bits 3..0 are security suite which is 0b0001 since Security Suite 1 is required</p> <p>Bit 4 is set to 0b1 since Authentication of the APDU is required.</p> <p>Bit 5 is set to 0b0 since the whole of an APDU is never encrypted</p> <p>Bit 6 is set to 0b0 since messages with MACs are unicast</p> <p>Bit 7 is set to 0b0 as per the Green Book</p>
	invocation counter (IC)	0x00000000	4	IC is always zero as specified in Section 8.4

1129 Table 7.2.5: Required components of MAC Header

1130 **7.2.6 Additional Authenticated Data (AAD) for the MAC
1131 calculation – informative**

1132 Terms in italics in this Section 7.2.6 shall have the meanings as specified in Green Book.

1133 The Green Book requires that the AAD used as input to the MAC calculation is the
1134 concatenation of:1135 SC || AK || transaction-id || originator-system-title || recipient-system-title || date-time ||
1136 other-information || information to be protected1137 The Green Book also requires that, for the elements contributing to AAD, only the values of
1138 the octet strings are included. The Green Book defines octet strings within the general-
1139 ciphering service in ASN.1 as:1140 *General-Ciphering ::= SEQUENCE*

1141 {

1142 <i>transaction-id</i>	OCTET STRING,
1143 <i>originator-system-title</i>	OCTET STRING,
1144 <i>recipient-system-title</i>	OCTET STRING,
1145 <i>date-time</i>	OCTET STRING,
1146 <i>other-information</i>	OCTET STRING,
1147 <i>key-info</i>	OPTIONAL,
1148 <i>ciphered-service</i>	OCTET STRING

1149 }

1150 As stated in Table 7.2.5, in GBCS-compliant APDUs:

- 1151 • SC takes the value 0x11; and
- 1152 • the following octet strings in the general-ciphering service shall have zero length and so
1153 have no value:
 - 1154 ○ transaction-id,
 - 1155 ○ originator-system-title,
 - 1156 ○ recipient-system-title,

- 1157 ○ date-time,
 1158 ○ other-information.

1159 As required by Section 4.3.3.4, AK is always absent.

1160 Thus, the AAD to be used in MAC calculations that protect APDUs is the concatenation:

1161 0x11 II information to be protected

1162 7.2.7 Message Construction - Grouping Header

1163 The following shall be the required components of the Grouping Header and shall be
 1164 populated with the values as per Table 7.2.7.

1165 Where a Signature is required in a message, it shall be calculated using only those attributes
 1166 marked 'Yes' in the 'Input to the ECDSA calculation' column of Table 7.2.7, in the sequence
 1167 they appear in the table.

1168 Thus, a KRP Signature or SMD Signature shall be calculated across the concatenation:

1169 Business Originator ID || Business Target ID || Originator Counter || date-time (if present)
 1170 || Message Code || Supplementary Remote Party ID (if present) || Supplementary
 1171 Remote Party Counter (if present) || Supplementary Originator Counter (if present) ||
 1172 Supplementary Remote Party Key Agreement Certificate (if present) || (information to be
 1173 protected)

1174 where (information to be protected) shall be:

- 1175 • the Command Payload in a Command;
- 1176 • the Response Payload in a Response; or
- 1177 • the Alert Payload in an Alert.

Grouping Header				
Input to the ECDSA calculation	xDLMS Message Elements	Contents	Length (octets)	Note
No	General-Signing	0xDF	1	xDLMS APDU tag for General-Signing (223 in decimal)
	transaction-id			
No	length	0x09	1	Length of Originator Counter plus 1
Yes	value	CRA Flag Originator Counter	9	CRA Flag shall be: 0x01 for Commands 0x02 for Responses 0x03 for Alerts
	originator-system-title			
No	length	0x08	1	Length of Entity Identifier
Yes	value	Business Originator ID	8	

Grouping Header				
Input to the ECDSA calculation	xDLMS Message Elements	Contents	Length (octets)	Note
	recipient-system-title			
No	length	0x08	1	Length of Entity Identifier
Yes	value	Business Target ID	8	
	date-time			
No	length	0x00 where no date / time is required in this Message 0x0C where a date / time field is required	1	Where date-time is not required for a Message, it shall be a 0 octet string as per the DLMS specification Where date-time is required for a Message, it shall be a 12 octet string as per the DLMS specification. See 'date-timestamp in response' column, 'Use Case reference' tab in Mapping Table
Yes	value	Either empty or a 12 character octet-string containing the date-time stamp for this Response	0 or 12	
	other-information			
No	length	Encoding(X)	variable Len(Encoding(X))	X is length of other information octet string. X is 2 or 18 or 26 or variable
Yes	value	Message Code Supplementary Remote Party ID Supplementary Remote Party Counter Supplementary Originator Counter Supplementary Remote Party Key Agreement Certificate	2 or 18 or 26 or variable	The Message Code shall always be present In an Alert, Supplementary Remote Party ID shall be present, if it is required by Section 16 In a Command or Response, the Supplementary Remote Party ID, Supplementary Remote Party Counter and Supplementary Originator Counter, shall be present or not in line with the

Grouping Header				
Input to the ECDSA calculation	xDLMS Message Elements	Contents	Length (octets)	Note
				requirements of Section 4.3.1.4 Supplementary Remote Party Key Agreement Certificate shall only be present where (1) this is a Command, (2) the Response to it should contain encrypted attributes and (3) the Supplementary Remote Party ID is for a Remote Party which does not already have a Key Agreement Public Key on the Device. It may only be present in Commands marked as allowing it in the column 'Key Agreement Certificate Potentially in Command?' of the Use Case reference tab of the Mapping Table
	Content			
No	length	Encoding(X)	Len(Encoding(X))	X is the length in octets of the Message Payload

Table 7.2.7: Required components of Grouping Header

1178 **7.2.8 Message Construction – ASN.1 Security Payloads**

1179 For Messages containing ASN.1 Security Payloads, the Payloads shall be constructed as
 1180 detailed in the Use Case for that Message Code (as defined by the Mapping Table).

1181 **7.2.9 Message Construction – DLMS COSEM Payloads**

1182 For Messages containing DLMS COSEM payloads (as defined by the Message Code and
 1183 Use Cases in Section 19):

- 1184 • any Command Payload shall comply with the requirements of Table 7.2.9a and the
 1185 associated Use Case;
- 1186 • any Response Payload shall comply with the requirements of Table 7.2.9b and the
 1187 associated Use Case; and
- 1188 • any Alert Payload shall comply with the requirements of Table 7.2.9c and the associated
 1189 Use Case.

DLMS COSEM Payloads – Commands				
No	xDLMS Message Elements	Contents	Length (octets)	Note
	access-request	0xD9	1	xDLMS APDU tag for Access Request (217 in decimal)
	long-invoke-id-and-priority	0x20 Least significant 24 bits of Originator Counter	4	Construction explained in rows below detailing bit (31..0) usage
	(bits 0-23) invoke-id	Least significant 24 bits of Originator Counter		
	(bits 24 -27) reserved	0b0000		Fixed value
	(bit 28) self-descriptive	0b0		Not-Self-Descriptive
	(bit 29) processing-option	0b1		Break on Error
	(bit 30) service-class	0b0		Unconfirmed
	(bit 31) priority	0b0		Normal
	date-time	0x00	1	A value for this element is not present so the length field is 0x00
	access-request-body			
	access-request-specification			
1	SEQUENCE OF	Use Case specific	1	<i>The total number of gets, sets and actions in the Use Case (means that there will be less than 128 in total). This content is specified in each DLMS COSEM Use Case</i>
2	Use Case Specific Content	Use Case specific		<i>The list of Gets, Sets and Actions specific to the Use Case. This content is specified in each DLMS COSEM Use Case</i>
	access-request-list-of-data			
	list-of-data			
3	SEQUENCE OF Data	Use Case specific	1	<i>The total number of attributes in the list-of-data in the Use Case (means that there will be less than 128 in total). This content is specified in each DLMS COSEM Use Case</i>
4	Use Case Specific Content	Use Case specific	Use Case set	<i>Values of the attributes required by the Use Case. This content is specified in each DLMS COSEM Use Case</i>

1191 Table 7.2.9a: Required components of Command Payload
 1192 Elements marked in Table 7.2.9a as Use Case specific shall be populated according to the
 1193 Use Case for the Message Code (see Section 19).

DLMS COSEM Payloads – Responses				
No	xDLMS Message Elements	Contents	Length (octets)	Note
	access-response	0xDA	1	xDLMS APDU tag for Access Response (218 in decimal)
	long-invoke-id-and-priority	0x20 Least significant 24 bits of Originator Counter	4	
	date-time	0x00	1	A value for this element is not needed so the length field is 0x00
	access-response-body			
	access-request - specification OPTIONAL	0x00	1	Not present so false (0x00)
	access-response-list-of-data			
	list-of-data			
5	SEQUENCE OF Data	Use Case specific	1	<i>The total number of attributes in the Response in the Use Case. This content is specified in each DLMS COSEM Use Case</i>
6	Use Case Specific Content	Use Case specific	Use Case set	<i>Values of the attributes required by the Use Case. This content is specified in each DLMS COSEM Use Case</i>
	access-response-specification			
7	SEQUENCE OF CHOICE	Use Case specific	1	<i>The total number of responses, including the 1 here and those in the Use Case</i>
8	Use Case Specific Content	Use Case specific	Use Case set	<i>Fields stating the result of each Gets, Sets and Actions specific to the Use Case.</i>

1194 Table 7.2.9b: Required components of Response Payload Elements marked in Table 7.2.9b as
 1195 Use Case specific shall be populated according to the Use Case for the Message Code (see
 1196 Section 19).

DLMS COSEM Payloads – Alerts				
No	xDLMS Message Elements	Contents	Length (octets)	Note
	data-notification	0x0F	1	xDLMS APDU tag for data-notification (15 in decimal)
	long-invoke-id-and-priority	0x20 least significant 24 bits of Originator	4	

DLMS COSEM Payloads – Alerts				
No	xDLMS Message Elements	Contents	Length (octets)	Note
		Counter		
	date-time	0x00	1	A value for this element is not needed so the length field is 0x00
	notification-body			
	structure	0x02	1	
1	SEQUENCE OF Data	0x02 unless there is Use Case specific data additional	1	<p><i>The majority of Alerts do not contain any additional data. For Alerts without additional data, there is no corresponding Use Case (since there is no Use Case specific content).</i></p> <p><i>Where an Alert does contain additional content, it has a specific Use Case. The additional content is specified in each such Use Case. In such cases, this field shall contain the total number of Data in the Use Case sequence plus the one in this template</i></p>
	Data			
	Tag	0x12	1	Tag for LONG UNSIGNED
	Value	Alert Code	2	The Alert Code for this Alert, shall be as defined in Section 16
	Data			
	Tag	0x09	1	Tag for octet-string
	Length	0x0C	1	Twelve characters long as DLMS date times are octet-string(12)
	Value	Time Stamp	12	The time stamp for this Alert, shall be as defined in Section 16
2	Use Case Specific Additional Content	Use Case specific	Use Case	<i>See Note at row 1, which means that, for most Alerts, there will be no Use Case specific content.</i>

1197 Table 7.2.9c: Required components of Alert Payload

1198 Elements marked in Table 7.2.9c as Use Case specific shall be populated according to the
1199 Use Case for the Message Code (see Section 19).1200

7.2.10 Message Construction – GBZ Payloads

1201 A GBZ Payload shall be a Payload containing one or more ZSE / ZCL commands. For
1202 clarity, this includes Payloads in HAN Only Commands between a PPMID and a GSME.

1203 For Messages containing GBZ Payloads (as defined by the Mapping Table):

- 1204 • any Command Payload shall comply with the requirements of Table 7.2.10a and the
1205 associated Use Case;
- 1206 • any Response Payload shall comply with the requirements of Table 7.2.10b and the
1207 associated Use Case; and

- 1208 • any Alert Payload shall comply with the requirements of Table 7.2.10c and the
 1209 associated Use Case.

1210 Each GBZ Use Case Specific Component shall comply with:

- 1211 • Table 7.2.10d if the ZSE / ZCL command within it is not encrypted; or
 1212 • Table 7.2.10e if the ZSE / ZCL command within it is encrypted.

GBZ Payloads – Commands

No	Message Elements	Contents	Length (octets)	Note
	Profile ID	0x0109	2	ZSE
1	<i>Total number of GBZ Use Case Specific Component(s)</i>	See 'Note' column	1	<i>This octet is to be interpreted as an 8 bit unsigned integer specifying the total number of GBZ Use Case Specific Component(s)</i>
2	<i>GBZ Use Case Specific Component(s)</i>	<i>Use Case specific</i>		<i>See Tables 7.2.10d and 7.2.10e</i>

1213 Table 7.2.10a: Required components of GBZ Command Payload Elements marked in Table
 1214 7.2.10a as Use Case specific shall be populated according to the Use Case for the Message
 1215 Code (see Section 15).

GBZ Payloads – Response

No	Message Elements	Contents	Length (octets)	Note
	Profile ID	0x0109	2	ZSE
1	<i>Total number of GBZ Use Case Specific Component(s)</i>	See 'Note' column	1	<i>This octet is to be interpreted as an 8 bit unsigned integer specifying the total number of GBZ Use Case Specific Component(s) in this Message</i>
2	<i>GBZ Use Case Specific Component(s)</i>	<i>Use Case specific</i>		<i>See Tables 7.2.10d and 7.2.10e</i>

1216 Table 7.2.10b: Required components of GBZ Response Payload Elements marked in Table
 1217 7.2.10b as Use Case specific shall be populated according to the Use Case for the Message
 1218 Code (see Section 15).

GBZ Payloads – Alerts

No	Message Elements	Contents	Length (octets)	Note
	Profile ID	0x0109	2	ZSE
1	<i>Total number of GBZ Use Case Specific Component(s)</i>	See 'Note' column	1	<i>This octet is to be interpreted as an 8 bit unsigned integer specifying the total number of GBZ Use Case Specific Component(s)</i>
	Alert Code	See 'Note' column	2	The Alert Code for this Alert as defined in Section 16
	Timestamp	UTCTime	4	The <i>UTCTime</i> , with its ZCL meaning, at which this Alert was created

GBZ Payloads – Alerts				
No	Message Elements	Contents	Length (octets)	Note
2	GBZ Use Case Specific Component(s)	Use Case specific		See Tables 7.2.10d and 7.2.10e

1219 Table 7.2.10c: Required components of GBZ Alert Payload

1220 Elements marked in Table 7.2.10c as Use Case specific shall be populated according to the
1221 Use Case for the Message Code (see Section 15).

GBZ Use Case Specific Component without encrypted content				
No	Message Elements	Contents	Length (octets)	Note
	Extended Header Control Field	0x00, 0x10, 0x11Or 0x01	1	Most significant nibble: 0x0 if ‘From Date Time’ not present; or 0x1 if ‘From Date Time’ present Least significant nibble: 0x0 (if not the last GBZ Use Case Specific Component in this Message) Or 0x1 (if the last GBZ Use Case Specific Component in this Message)
	Extended Header Cluster ID	See ‘Note’ column	2	The Cluster ID of the ZSE / ZCL command contained in this GBZ Use Case Specific Component
	Extended Header GBZ Command Length	See ‘Note’ column	2	These two octets shall be interpreted as a 16 bit unsigned integer specifying the total length on octets of the remainder of this GBZ Component (so excluding this and prior fields)
	From Date Time	See ‘Note’ column	4	The earliest date-time of any entry that can be returned in the response, specified as a ZigBee UTCTime.
	ZCL header	Use Case specific	3	These fields shall have the meaning specified in the ZSE / ZCL Specifications except for the Transaction Sequence Number. The Transaction Sequence Number shall be set to 0 for the first request-style ZSE / ZCL command in the Message and shall be incremented by one for every subsequent request-style ZSE / ZCL command frame in the Message. The corresponding response-style ZSE / ZCL command frame shall copy the Transaction Sequence Number from the request-style ZSE / ZCL command frame
	ZCL payload	Use Case specific	Variable	These fields shall have the meaning specified in the ZSE / ZCL specifications

1222

Table 7.2.10d: Required components of GBZ Use Case Specific Component without encrypted content

GBZ Use Case Specific Component with encrypted content				
No	Message Elements	Contents	Length (octets)	Note
	Extended Header Control Field	0x02 Or 0x03	1	0x02 (if not the last GBZ Use Case Specific Component in this Message) Or 0x03 (if the last GBZ Use Case Specific Component in this Message)
	Extended Header Cluster ID	See 'Note' column	2	The Cluster ID of the ZCL Command contained in this GBZ Use Case Specific Component
	Extended Header GBZ Command Length	See 'Note' column	2	These two octets shall be interpreted as a 16 bit unsigned integer specifying the total length in octets of the remainder of this GBZ Component (so excluding this and prior fields but including the 2 octets of Additional Header)
	Additional Header Control	0x00	1	Reserved for future extensibility
	Additional Header Frame Counter	See 'Note' column	1	This octet is to be interpreted as an 8 bit unsigned integer. Its value shall be 0x00 for the first GBZ Use Case Specific Component with encrypted content in a Message. The value shall increase by one in each subsequent GBZ Use Case Specific Component with encrypted content in a Message
	ZCL header	See 'Note' column	3	These fields shall have the meaning specified in the ZigBee Cluster Library
	Length of Ciphered Information	See 'Note' column	2	These two octets shall be interpreted as a 16 bit unsigned integer specifying the total length in octets of the Ciphered Information
	Ciphered Information	See 'Note' column	Variable	See Section 8.4

Table 7.2.10e: Required components of GBZ Use Case Specific Component with encrypted content

1223

7.2.11 Transfer of Large Remote Party Messages

1224
1225

All Devices which are not Type 2 Devices shall be capable of supporting the General Block Transfer (GBT) requirements of this Section 7.2.11.

1226

7.2.11.1 GBT Terminology and Parameters¹⁸

1227
1228

A GBT Message shall be an APDU constructed and processed as defined by this Section 7.2.11.

1229
1230

A GBT Message Series shall be the set of GBT Messages needed to exchange one complete Remote Party Message between a GBT Initiator and a GBT Recipient.

¹⁸ Terms defined within this section are only used within this section, and therefore not included in the Glossary (Section 21).

1231 For a Remote Party Command sent using a GBT Message Series, the GBT Initiator shall be
1232 the Access Control Broker and the GBT Recipient shall be the target Device.

1233 For a Remote Party Response or a Remote Party Alert sent using a GBT Message Series,
1234 the GBT Initiator shall be the sending Device and the GBT Recipient shall be the Access
1235 Control Broker.

1236 A GBT Third Party shall be the Remote Party identified by:

1237 • in a Remote Party Command, the value in the Business Originator ID field; and

1238 • in a Remote Party Response or a Remote Party Alert, the value in the Business Target
1239 ID field.

1240 GBT Streaming Window shall be the number of GBT Messages the GBT Initiator sends
1241 without receipt of a GBT Message (Acknowledgement), or since receipt of the most recent
1242 GBT Message (Acknowledgement).

1243 GBT Streaming Window shall be:

1244 • 63 where the GBT Message Series carries a Remote Party Response;

1245 • 6 where the GBT Message Series carries a Remote Party Command; or

1246 • the number of GBT Messages the sender wishes to be resent in response to a GBT
1247 Message (Request Block Resend).

1248 Maximum PDU Size shall be 1200 octets.

7.2.11.2 Remote Party Message size

1250 Where a Remote Party Message exceeds the Maximum PDU Size, the GBT Initiator and the
1251 GBT Responder shall exchange the Remote Party Message in a GBT Message Series.

1252 Where a Remote Party Message does not exceed the Maximum PDU Size, the GBT Initiator
1253 and the GBT Responder may exchange the Remote Party Message in a GBT Message
1254 Series.

7.2.11.3 GBT Message Structure

1256 A GBT Message shall, if it is a GBT Message (Acknowledgement) or a GBT Message
1257 (Request Block Resend), be the concatenation:

1258 Message Routing Header || GBT Header

1259 A GBT Message shall, if it is neither a GBT Message (Acknowledgement) nor a GBT
1260 Message (Request Block Resend), be the concatenation:

1261 Message Routing Header || GBT Header || GBT Block Data

1262 where:

- 1263 • Message Routing Header shall be structured and populated according to Table 7.2.11.5.
1264 Note that Message Routing Header (1) uniquely identifies the GBT Message Series, (2)
1265 identifies whether the GBT Message is being sent to or from the Device and (3)
1266 unambiguously ties all GBT Messages in the GBT Message Series to the single Remote
1267 Party Message being exchanged;
- 1268 • GBT Header shall be structured and populated according to Table 7.2.11.6; and
- 1269 • GBT Block Data shall be the part of the Remote Party Message, constructed as per
1270 Section 7.2.11.4, being carried in this GBT Message.

1271 7.2.11.4 GBT Message processing

1272 The GBT Initiator shall, once the Remote Party Message is fully constructed and all
1273 cryptographic protections are applied, slice the octet string produced so that:

- 1274 9. GBT Block Data with GBT Initiator Block Number of 1 is the 1149 most significant octets
1275 of the Remote Party Message, or all of the octets if the size of the Remote Party
1276 Message is less than 1149 bytes;
- 1277 10. GBT Block Data with GBT Initiator Block Number of 2 is the next 1149 most significant
1278 octets of the Remote Party Message, or all of the octets if the size of the remaining
1279 octets in Remote Party Message is less than 1149 bytes; and
- 1280 11. remaining GBT Block Data are created by repeating Step 2, each time incrementing GBT
1281 Initiator Block Number by 1, until there are no remaining octets in the Remote Party
1282 Message.

1283 The GBT Recipient shall not undertake any processing, in the sense of Section 6, of the
1284 Remote Party Message carried in a GBT Message Series until it has received:

- 1285 • a GBT Message in this GBT Message Series where the 'last-block' field contains 0b1
1286 (meaning last block); and
- 1287 • all GBT Messages in this GBT Message Series with 'block-number' fields less than the
1288 'block-number' field in the last block. Where the GBT Recipient has not received all
1289 such GBT Messages, it shall send a GBT Message (Request Block Resend) for each
1290 missing block-number. Where the GBT Recipient is a Device, it may discard all blocks
1291 in a GBT Message Series if it has received no response to a GBT Message (Request
1292 Block Resend) after 60 minutes.

1293 When a GBT Recipient receives a GBT Message with 'block-number' being an integer
1294 multiple of GBT Streaming Window for this GBT Message Series, it shall send a GBT
1295 Message (Acknowledgement).

1296 GBT Recipient Block Number shall be set to 0x0001 in the first GBT Message sent by the
1297 GBT Recipient. It shall be incremented by 1 in each subsequent GBT Message it sends.

1298 GBT Initiator Block Number Ack shall be the highest of:

- 1299 • 0x0000; and
- 1300 • the highest block-number in any GBT Message the GBT Initiator has received in this
1301 GBT Message Series.

1302 GBT Recipient Block Number Ack shall:

- 1303 • in a GBT Message (Acknowledgement), be the highest block-number in any GBT
1304 Message the GBT Recipient has received in this GBT Message Series; and
- 1305 • in a GBT Message (Request Block Resend), the value of block-number up to which the
1306 GBT Recipient has received all the prior numbered GBT Messages in this GBT
1307 Message Series.

1308 Where the GBT Initiator is a Device, the Device shall be able to resend any GBT Message
1309 within a GBT Message Series, for a minimum period from when it sends the first GBT
1310 Message in that series, to whichever is the sooner of:

- 1311 • it receiving an authenticated GBT Message (Acknowledgement) where the GBT
1312 Recipient Block Number Ack contains a value equal to the highest value of GBT Initiator
1313 Block Number Ack in this GBT Message Series; or
- 1314 • 24 hours later.

7.2.11.5 Message Routing Header

Message Routing Header				
No	xDLMS Message Elements	Contents	Length (octets)	Note
	general-ciphering	0xDD	1	Tag used is the same as for a normal DLMS General-Ciphering header
	transaction-id			
	Length	0x09	1	Length of Originator Counter
	Value	See 'Note' column	9	Shall be populated with the corresponding field from the Grouping Header in the Remote Party Message that is being carried in this GBT Message Series
	originator-system-title			
	Length	0x08	1	Length of Entity Identifier
	Value	Business Originator ID	8	If the GBT Message is sent from the Device, the value shall be the Entity Identifier of the Device. If the GBT Message is sent to the Device, the value shall be the Entity Identifier of the GBT Third Party
	recipient-system-title			
	Length	0x08	1	Length of Entity Identifier
	Value	Business Target ID	8	If the GBT Message is sent to the Device, the value shall be the Entity Identifier of the Device. If the GBT Message is sent from the Device, the value shall be the Entity Identifier of the GBT Third Party.
	date-time	0x00	1	A value for this element is not needed so the length field is 0x00
	other-information			
	Length	0x02	1	Length of Message Code
	Value	Message Code	2	Shall be populated with the corresponding field from the Grouping Header in the Remote Party Message that is being carried in this GBT Message Series
	key-info	0x00	1	key-info values are not present so encoded as 0x00
	ciphered-service			
	Length	Encoding(X)	Len(Encoding(X))	X shall be the length in octets of the subsequent parts of the GBT

Message Routing Header				
No	xDLMS Message Elements	Contents	Length (octets)	Note
				Message after this length value.
	security header			
	security control byte (SC)	0x01	1	Specifies that no MAC field is present at the end of the APDU
	invocation counter (IC)	0x00000000	4	IC is always zero

1316 Table 7.2.11.5: Message Routing Header

1317 **7.2.11.6 GBT Header**

GBT Header				
No	xDLMS Message Elements	Contents	Length (octets)	Note
	general-block-transfer	0xE0	1	xDLMS APDU tag for General-Block-Transfer
	block-control			
	last-block (bit 7)	See 'Note' column	1/8	0b0 if not the last GBT Message in this GBT Message Series the sender has to send, or 0b1 if this is the last GBT Message in this GBT Message Series the sender has to send
	streaming (bit 6)	See 'Note' column	1/8	0b1 if the sender does not require a GBT Message in response, or 0b0 if the sender does require a GBT Message in response
	window (bits 5 – 0)	See 'Note' column	5/8	The value of GBT Streaming Window as required by Section 7.2.11.1.
	block-number	See 'Note' column	2	GBT Initiator Block Number, if this GBS Message is sent by the GBT Initiator. GBT Recipient Block Number, if this GBS Message is sent by the GBT Recipient.
	block-number-ack	See 'Note' column	2	GBT Initiator Block Number Ack, if this GBS Message is sent by the GBT Initiator. GBT Recipient Block Number Ack, if this GBS Message is sent by the GBT Recipient.
	block-data			
	Length	Encoding(X)	Len(Encoding(X))	X is the length in octets of the following parts of this APDU.

1318 Table 7.2.11.6: GBT Header

1319 **7.2.11.7 Illustrations – informative**

1320 GBT allows for the transport of Messages where the Message is greater than the Maximum
1321 PDU Size. A number of Use Cases can result in this larger Message size, either as a
1322 Command or a Response. There are no Alert Use Cases that result in the larger Message
1323 size.

1324 GBT does not change any part of the Remote Party Message content that is being
1325 transported.

1326 Example 1: A small Command with small Response – e.g. read MPAN.

1327 Without GBT, the Command is:

1328 MAC Header || Grouping Header || read MPAN Command Payload || 0x00 || ACB-SMD
1329 MAC

1330 and the Response is:

1331 MAC Header || Grouping Header || read MPAN Response Payload || 0x00 || SMD-KRP
1332 MAC

1333 GBT can be applied to this Use Case. The Command becomes:

1334 Message Routing Header || GBT Header || MAC Header || Grouping Header || read
1335 MPAN Command Payload || 0x00 || ACB-SMD MAC

1336 and the Response becomes:

1337 Message Routing Header || GBT Header || MAC Header || Grouping Header || read
1338 MPAN Response Payload || 0x00 || SMD-KRP MAC

1339 Example 2: A large Command with small Response – e.g. set tariff on an ESME where the
1340 tariff is complex.

1341 Without GBT, the Command is:

1342 MAC Header || Grouping Header || set Tariff Command Payload || 0x40 || KRP
1343 Signature || ACB-SMD MAC

1344 For the purposes of example, assume this is divided into three blocks, so block-numbers 1, 2
1345 and 3. Actual set tariff Commands will vary from this number of blocks.

1346 The Command is transmitted as the GBT Message Series:

1347 Message Routing Header || GBT Header || block 1
1348 Message Routing Header || GBT Header || block 2
1349 Message Routing Header || GBT Header || block 3

1350 This is reconstructed at the ESME, by concatenating blocks 1, 2 and 3 to give:

1351 MAC Header || Grouping Header || set Tariff Command Payload || 0x40 || KRP
1352 Signature || ACB-SMD MAC

1353 The Response would have the following structure:

1354 Message Routing Header || GBT Header || Grouping Header || Response Payload ||
1355 0x40 || SMD Signature

1356 It will be smaller than the Maximum PDU Size and so can be sent as a single APDU without
1357 any use of GBT.

1358 Example 3: small Command with large Response – e.g. read half-hourly profile (Export)

1359 The Command is:

1360 MAC Header || Grouping Header || read half-hourly profile (Export) Command Payload ||
1361 0x00 || ACB-SMD MAC

1362 It will be smaller than the Maximum PDU Size and so can be sent as a single APDU without
1363 any use of GBT.

1364 The ESME Response is:

1365 MAC Header || Grouping Header || read half-hourly profile (Export) Response Payload ||
1366 0x00 || SMD-KRP MAC

1367 Assuming that there are 75 blocks to send the GBT Message Series would, if no retries are
1368 needed, be as follows:

1369 The ESME will send:

1370 Message Routing Header || GBT Header || Block 1

1371 ...

1372 Message Routing Header || GBT Header || Block 63

1373 and wait for acknowledgement.

1374 The Access Control Broker will construct and send that acknowledgement whose structure is:

1375 Message Routing Header || GBT Header

1376 When this acknowledgement is received by the ESME, the ESME will send:

1377 Message Routing Header || GBT Header || Block 64

1378 ...

1379 Message Routing Header || GBT Header || Block 75

1380 When the whole Message is received by the Access Control Broker, the Response can then
1381 be reconstructed:

1382 MAC Header || Grouping Header || read half-hourly profile (Export) Response Payload ||
1383 0x00 || SMD-KRP MAC

1384 Once the response has been reconstructed, the MAC can be checked.

1385 **7.3 Device Requirements – DLMS COSEM**

1386 **7.3.1 Introduction – informative**

1387 The DLMS COSEM server in the ESME (and CHF where a DLMS COSEM server is present)
1388 responds to requests for information, and also provides Alerts in response to events within
1389 the meter (e.g. push data at the end of billing period; Alert in the event of a tamper; disable
1390 supply when prepayment credit expires). To achieve this, a level of configuration is needed
1391 to ensure that the behaviour of the Device is as expected.

1392 SMETS and CHTS require that the Critical Commands mandated by them (and so those
1393 defined in the GBCS) are the only Critical commands allowed. Devices may implement
1394 additional non Critical features only.

1395 SMETS and CHTS only require DLMS COSEM support on the ESME.

1396 DLMS COSEM objects (or functionality equivalent to them) are required to deliver the ESME
1397 functionality defined in the Use Cases in a consistent way but should not be accessible via
1398 the ESME's HAN interface (i.e. it is internal functionality).

1399 **7.3.2 General Requirements**

1400 Constant values specified in Table 7.3.8a shall be fixed before operation and shall be
1401 immutable save via a firmware upgrade. This is to ensure consistent functioning and guard

1402 against potential attacks. Except where explicitly required by this Section 7.3, a Device shall
1403 not expose any part of any DLMS COSEM object, either for the writing of an attribute or for
1404 the invocation of a method that could, if used, constitute a Critical action.

1405 For Devices which are not ESME or CHF (so where `deviceType <> 1 or 2`), the GBCS
1406 does not require the implementation of any DLMS COSEM objects.

1407 All Devices which are ESME (so where `deviceType = 1`):

- 1408 • shall implement all of the DLMS COSEM objects, attributes and methods detailed in
1409 'SMETS Required Objects' tab of the table in Section 20, and expose the specified
1410 attributes and methods over its network interface; and
- 1411 • shall have the constant values set for the DLMS COSEM attributes specified as
1412 requiring constant values in Table 7.3.8a, and shall ensure that such values cannot be
1413 amended, save via activation of new firmware.

7.3.3 Application Associations

1415 Any ESME or CHF shall communicate using pre-established Application Associations (AA).
1416 These shall be set at manufacture, and the Device shall reject all subsequent attempts to
1417 open or release Application Associations.

1418 An ESME or CHF shall support the Application Associations in Table 7.3.8c. An ESME or
1419 CHF shall not support any additional Application Associations.

1420 The Application Associations in Table 7.3.8c shall limit access to DLMS features by
1421 configuring the `object_list` attribute to reflect the access granted to the role in 'SMETS
1422 Required Objects' tab of the table in Section 20. Any other methods and attributes of any
1423 class shall be made inaccessible by listing them in the `object_list` attribute such that there is
1424 no access.

1425 The Public AA shall only expose:

- 1426 • the SAP Assignment object; and
- 1427 • the DLMS COSEM Logical Device name object and `object_list` with no objects listed
1428 other than the Association Logical Name (LN) Object (with its Blue Book meaning) and
1429 the SAP Assignment Object.

1430 When a Message is received by the ESME or CHF, the Message shall be validated against
1431 the AA based on the Business Originator ID and the Message Code within the Grouping
1432 Header.

1433 Other attributes in the Association LN Objects and the Security Setup Objects shall be set at
1434 manufacture in accordance with Tables 7.3.8d and 7.3.8e.

1435 The 'SAP Assignment' object shall be configured at manufacture in accordance with Table
1436 7.3.8f. The method associated with the 'SAP Assignment' object shall not be accessible to
1437 any Application Association.

7.3.4 Interface Classes and Objects

1439 Devices shall support the version of Interface Classes shown as current in the Blue Book.

1440 An ESME shall support the 'Class 9000' as detailed in Section 22 of this GBCS.

1441 Unless explicitly required in a predetermined script or the SMETS 'Required Objects' tab of
1442 the Mapping Table, Class 3 objects shall not have a reset method that is accessible external
1443 to the Device.

1444 Unless otherwise stated, Generic Profile objects with a non-zero attribute 4 shall capture the
1445 first entry at midnight UTC.

1446 The ESME shall have the constant values set for the DLMS COSEM attributes specified as
1447 requiring constant values in Table 7.3.8a, and shall ensure that such values cannot be
1448 amended, save via activation of new firmware.

1449 **7.3.5 Values normally negotiated when an AA is established**

1450 **7.3.5.1 Conformance Block Contents**

1451 The conformance block shall be set according to Table 7.3.8g.

1452 **7.3.5.2 Other Items.**

1453 Other items for pre-establishing the Application Associations and other communication
1454 parameters shall be implemented as detailed in Table 7.3.8h.

1455 **7.3.5.3 Security Setup Objects**

1456 Security Setup Objects shall be limited to those listed in Table 7.3.8c.

1457 Manufacturer specific attributes and methods for these objects shall not be accessible
1458 external to the Device.

1459 The methods of the Security Setup objects shall not be accessible external to the Device.
1460 The attributes of the Security Setup objects shall be as specified in Table 7.3.8e.

1461 Note that Security Credentials are updated as specified in Section 13.

1462 **7.3.6 Scripts for operation of the meter**

1463 Scripts required for operation of the Device shall be as listed in Table 7.3.8b.

1464 The Device shall ensure that the script table objects shall be read only. The Device shall
1465 ensure that a script table object entries shall only be executable by the corresponding
1466 Application Association specified in Table 7.3.8b.

1467 The Device shall ensure that a script table object's entries shall only be executable from an
1468 activity calendar, scheduler, or single action scheduler controlled by the corresponding
1469 Application Association application in Table 7.3.8b.

1470 **7.3.7 Pricing matrices, scripts and registers**

1471 **7.3.7.1 Summary of approach - informative**

1472 As required by SMETS, an ESME has:

- 1473 • a 1 * 48 matrix of primary element TOU (Time Of Use) consumption registers, and an
1474 associated 1*48 matrix of consumption based prices;
- 1475 • a 4 block by 8 time band matrix of primary element 'TOU with Blocks' consumption
1476 registers, and an associated 4 * 8 matrix of consumption based prices;
- 1477 • a tariff switching table, which specifies time based switching between primary
1478 consumption registers and so between different prices. (Switching between blocks is
1479 based on consumption passing thresholds and not on time);
- 1480 • for twin element ESME only, a 1 * 4 matrix of secondary element TOU consumption
1481 registers, and an associated 1*4 matrix of consumption based prices; and
- 1482 • for twin element ESME only, a secondary tariff switching table, which specifies time
1483 based switching between registers and so between different consumption based prices.

1484 There needs to be a clear mapping, at the level of encoded instructions in Commands to the
1485 ESME, between the switching table entries (identified by Script Selector), Consumption
1486 Registers and consumption based prices (identified by Index). The next section details that
1487 mapping for the encoded DLMS COSEM elements used by the ESME.

1488 Note that the mapping of Script Selector is included for information in Table 7.3.7.2, but the
 1489 requirements for that mapping are in Section 7.3.8.

1490 **7.3.7.2 ESME requirements**

1491 An ESME shall:

- 1492 • in calculating the cost of Consumption, for a Consumption Register specified in Table
 1493 7.3.7.2 apply the charge_per_unit in the charge_table_element identified by
 1494 corresponding Index specified in Table 7.3.7.2 of the unit_charge_active attribute of the
 1495 corresponding Charge Object specified in Table 7.3.7.2; and
- 1496 • reject any instruction to set the unit_charge_passive attribute of a Charge Object in
 1497 Table 7.3.7.2 that does not include all of the charge_table_elements specified in Table
 1498 7.3.7.2 for that Charge Object or contains charge_table_elements with values of index
 1499 that are not in Table 7.3.7.2 for that Charge Object.

Description	Script Selector (long unsigned)	Consumption Register	Charge Object	Index (octet-string(1))	Notes
TOU(1)	0x0001	1-0:1.8.1.255	0-0:19.2.0.255	0x01	
TOU(2)	0x0002	1-0:1.8.2.255	0-0:19.2.0.255	0x02	
TOU(3)	0x0003	1-0:1.8.3.255	0-0:19.2.0.255	0x03	
TOU(4)	0x0004	1-0:1.8.4.255	0-0:19.2.0.255	0x04	
TOU(5)	0x0005	1-0:1.8.5.255	0-0:19.2.0.255	0x05	
TOU(6)	0x0006	1-0:1.8.6.255	0-0:19.2.0.255	0x06	
TOU(7)	0x0007	1-0:1.8.7.255	0-0:19.2.0.255	0x07	
TOU(8)	0x0008	1-0:1.8.8.255	0-0:19.2.0.255	0x08	
TOU(9)	0x0009	1-0:1.8.9.255	0-0:19.2.0.255	0x09	
TOU(10)	0x000A	1-0:1.8.10.255	0-0:19.2.0.255	0x0A	
TOU(11)	0x000B	1-0:1.8.11.255	0-0:19.2.0.255	0x0B	
TOU(12)	0x000C	1-0:1.8.12.255	0-0:19.2.0.255	0x0C	
TOU(13)	0x000D	1-0:1.8.13.255	0-0:19.2.0.255	0x0D	
TOU(14)	0x000E	1-0:1.8.14.255	0-0:19.2.0.255	0x0E	
TOU(15)	0x000F	1-0:1.8.15.255	0-0:19.2.0.255	0x0F	
TOU(16)	0x0010	1-0:1.8.16.255	0-0:19.2.0.255	0x10	
TOU(17)	0x0011	1-0:1.8.17.255	0-0:19.2.0.255	0x11	
TOU(18)	0x0012	1-0:1.8.18.255	0-0:19.2.0.255	0x12	
TOU(19)	0x0013	1-0:1.8.19.255	0-0:19.2.0.255	0x13	
TOU(20)	0x0014	1-0:1.8.20.255	0-0:19.2.0.255	0x14	
TOU(21)	0x0015	1-0:1.8.21.255	0-0:19.2.0.255	0x15	
TOU(22)	0x0016	1-0:1.8.22.255	0-0:19.2.0.255	0x16	
TOU(23)	0x0017	1-0:1.8.23.255	0-0:19.2.0.255	0x17	
TOU(24)	0x0018	1-0:1.8.24.255	0-0:19.2.0.255	0x18	
TOU(25)	0x0019	1-0:1.8.25.255	0-0:19.2.0.255	0x19	
TOU(26)	0x001A	1-0:1.8.26.255	0-0:19.2.0.255	0x1A	
TOU(27)	0x001B	1-0:1.8.27.255	0-0:19.2.0.255	0x1B	

Description	Script Selector (long unsigned)	Consumption Register	Charge Object	Index (octet-string(1))	Notes
TOU(28)	0x001C	1-0:1.8.28.255	0-0:19.2.0.255	0x1C	
TOU(29)	0x001D	1-0:1.8.29.255	0-0:19.2.0.255	0x1D	
TOU(30)	0x001E	1-0:1.8.30.255	0-0:19.2.0.255	0x1E	
TOU(31)	0x001F	1-0:1.8.31.255	0-0:19.2.0.255	0x1F	
TOU(32)	0x0020	1-0:1.8.32.255	0-0:19.2.0.255	0x20	
TOU(33)	0x0021	1-0:1.8.33.255	0-0:19.2.0.255	0x21	
TOU(34)	0x0022	1-0:1.8.34.255	0-0:19.2.0.255	0x22	
TOU(35)	0x0023	1-0:1.8.35.255	0-0:19.2.0.255	0x23	
TOU(36)	0x0024	1-0:1.8.36.255	0-0:19.2.0.255	0x24	
TOU(37)	0x0025	1-0:1.8.37.255	0-0:19.2.0.255	0x25	
TOU(38)	0x0026	1-0:1.8.38.255	0-0:19.2.0.255	0x26	
TOU(39)	0x0027	1-0:1.8.39.255	0-0:19.2.0.255	0x27	
TOU(40)	0x0028	1-0:1.8.40.255	0-0:19.2.0.255	0x28	
TOU(41)	0x0029	1-0:1.8.41.255	0-0:19.2.0.255	0x29	
TOU(42)	0x002A	1-0:1.8.42.255	0-0:19.2.0.255	0x2A	
TOU(43)	0x002B	1-0:1.8.43.255	0-0:19.2.0.255	0x2B	
TOU(44)	0x002C	1-0:1.8.44.255	0-0:19.2.0.255	0x2C	
TOU(45)	0x002D	1-0:1.8.45.255	0-0:19.2.0.255	0x2D	
TOU(46)	0x002E	1-0:1.8.46.255	0-0:19.2.0.255	0x2E	
TOU(47)	0x002F	1-0:1.8.47.255	0-0:19.2.0.255	0x2F	
TOU(48)	0x0030	1-0:1.8.48.255	0-0:19.2.0.255	0x30	
Block(1)TOU(1)	0x0065	1-1:1.8.1.255	0-0:19.2.0.255	0xA1	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(1)TOU(2)	0x0066	1-1:1.8.2.255	0-0:19.2.0.255	0xA2	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(1)TOU(3)	0x0067	1-1:1.8.3.255	0-0:19.2.0.255	0xA3	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(1)TOU(4)	0x0068	1-1:1.8.4.255	0-0:19.2.0.255	0xA4	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(1)TOU(5)	0x0069	1-1:1.8.5.255	0-0:19.2.0.255	0xA5	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(1)TOU(6)	0x006A	1-1:1.8.6.255	0-0:19.2.0.255	0xA6	Which block is activated by this Script Selector will depend on ESME consumption since last

Description	Script Selector (long unsigned)	Consumption Register	Charge Object	Index (octet-string(1))	Notes
					block reset
Block(1)TOU(7)	0x006B	1-1:1.8.7.255	0-0:19.2.0.255	0xA7	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(1)TOU(8)	0x006C	1-1:1.8.8.255	0-0:19.2.0.255	0xA8	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(2)TOU(1)	0x006D	1-2:1.8.1.255	0-0:19.2.0.255	0xB1	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(2)TOU(2)	0x006E	1-2:1.8.2.255	0-0:19.2.0.255	0xB2	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(2)TOU(3)	0x006F	1-2:1.8.3.255	0-0:19.2.0.255	0xB3	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(2)TOU(4)	0x0070	1-2:1.8.4.255	0-0:19.2.0.255	0xB4	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(2)TOU(5)	0x0071	1-2:1.8.5.255	0-0:19.2.0.255	0xB5	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(2)TOU(6)	0x0072	1-2:1.8.6.255	0-0:19.2.0.255	0xB6	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(2)TOU(7)	0x0073	1-2:1.8.7.255	0-0:19.2.0.255	0xB7	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(2)TOU(8)	0x0074	1-2:1.8.8.255	0-0:19.2.0.255	0xB8	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(3)TOU(1)	0x0075	1-3:1.8.1.255	0-0:19.2.0.255	0xC1	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(3)TOU(2)	0x0076	1-3:1.8.2.255	0-0:19.2.0.255	0xC2	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(3)TOU(3)	0x0077	1-3:1.8.3.255	0-0:19.2.0.255	0xC3	Which block is activated by this Script Selector will depend on ESME consumption since last block reset

Description	Script Selector (long unsigned)	Consumption Register	Charge Object	Index (octet-string(1))	Notes
Block(3)TOU(4)	0x0078	1-3:1.8.4.255	0-0:19.2.0.255	0xC4	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(3)TOU(5)	0x0079	1-3:1.8.5.255	0-0:19.2.0.255	0xC5	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(3)TOU(6)	0x007A	1-3:1.8.6.255	0-0:19.2.0.255	0xC6	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(3)TOU(7)	0x007B	1-3:1.8.7.255	0-0:19.2.0.255	0xC7	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(3)TOU(8)	0x007C	1-3:1.8.8.255	0-0:19.2.0.255	0xC8	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(4)TOU(1)	0x007D	1-4:1.8.1.255	0-0:19.2.0.255	0xD1	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(4)TOU(2)	0x007E	1-4:1.8.2.255	0-0:19.2.0.255	0xD2	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(4)TOU(3)	0x007F	1-4:1.8.3.255	0-0:19.2.0.255	0xD3	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(4)TOU(4)	0x0080	1-4:1.8.4.255	0-0:19.2.0.255	0xD4	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(4)TOU(5)	0x0081	1-4:1.8.5.255	0-0:19.2.0.255	0xD5	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(4)TOU(6)	0x0082	1-4:1.8.6.255	0-0:19.2.0.255	0xD6	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(4)TOU(7)	0x0083	1-4:1.8.7.255	0-0:19.2.0.255	0xD7	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(4)TOU(8)	0x0084	1-4:1.8.8.255	0-0:19.2.0.255	0xD8	Which block is activated by this Script Selector will depend on ESME consumption since last block reset

Description	Script Selector (long unsigned)	Consumption Register	Charge Object	Index (octet-string(1))	Notes
TOU(1) (Secondary Element)	0x00C9	1-20:1.8.1.255	0-0:19.2.5.255	0x01	Only present on twin element ESME
TOU(2) (Secondary Element)	0x00CA	1-20:1.8.2.255	0-0:19.2.5.255	0x02	Only present on twin element ESME
TOU(3) (Secondary Element)	0x00CB	1-20:1.8.3.255	0-0:19.2.5.255	0x03	Only present on twin element ESME
TOU(4) (Secondary Element)	0x00CC	1-20:1.8.4.255	0-0:19.2.5.255	0x04	Only present on twin element ESME

Table 7.3.7.2: ESME requirements for pricing matrices, scripts and registers

7.3.8 DLMS Device Requirements Tables

- 1500
1501 Table 7.3.8a: Objects tab in embedded file
1502 Table 7.3.8b: Scripts tab in embedded file
1503 Table 7.3.8c: Application Associations tab in embedded file
1504 Table 7.3.8d: Association LN Object Content tab in embedded file
1505 Table 7.3.8e: Security Setup Object Content tab in embedded file
1506 Table 7.3.8f: SAP Assignment Object content tab in embedded file
1507 Table 7.3.8g: Conformance Content tab in embedded file
1508 Table 7.3.8h: End to End Communications tab in embedded file



GBCS v0.8.1 DLMS
Device Requirements.:

7.3.9 ESME accounts, credits and charges - informative

1511 As detailed in the Mapping Table, an ESME shall have two Account objects (Class ID 111)
1512 which shall be used in both Credit and Prepayment Modes:

- 1513 • a single 'active' Account object (OBIS code 0-0:19.0.0.255) which can be read in Use
1514 Cases but which is never written to directly; and
- 1515 • a single 'passive' Account object (OBIS code 0-1:19.0.0.255) which can be written to in
1516 Use Cases but which is never read.

1517 Both relate to Import Energy.

1518 The activation attribute and method of the 'passive' object leads to its static values being
1519 copied from the passive to the active object, rather than the passive becoming active (see
1520 Section 9.2.2.7).

1521 The 'Set Payment Mode to Credit' and 'Set Payment Mode to Prepayment' Use Cases are
1522 used to trigger such activation of the 'passive' object. The SMETS attributes Suspend Debt
1523 Emergency and Suspend Debt Disabled are implemented as part of these objects and so
1524 are set in these Use Cases. If an ESME is in Prepayment Mode and the value of either

1525 Suspend Debt attribute is to be changed, this can be achieved by sending a Set Payment
 1526 Mode to Prepayment Command containing the new values.

7.3.10 ESME requirements for using Special Days objects

1528 When applying Blue Book special days related requirements to a Calendar / Scheduler object listed in
 1529 Table 7.3.10, an ESME shall use the corresponding Specials Days Object in Table 7.3.10.

<i>SMETS Reference</i>	<i>Calendar / Scheduler object</i>		<i>Special Days Object to be used</i>
	<i>Class ID</i>	<i>OBIS</i>	<i>OBIS</i>
TariffSwitchingTable(SpecialDays)	20	0-0:13.0.0.255	0-0:11.0.0.255
TariffSwitchingTable(SecondaryElement)(SpecialDays)	20	0-0:13.0.1.255	0-0:11.0.1.255
Non-DisablementCalendar(SpecialDays)	10	0-0:12.0.1.255	0-0:11.0.2.255
AuxiliaryLoadControlSwitchesCalendar(SpecialDays)	10	0-0:12.0.2.255	0-0:11.0.3.255

1530 Table 7.3.10: Special Days Object

7.4 Device requirements – ZSE

1532 This Section 7.4 details the ZigBee clusters, attributes and commands that shall be
 1533 supported by Devices in their interactions with other Devices on the same HAN, including
 1534 whether the support is as a ZSE client or a server. Note, this section does not detail the ZCL
 1535 / ZSE commands that Devices will need to process as part of processing Remote Party
 1536 Commands, or Commands sent by a PPMID to a GSME. Such requirements are detailed in
 1537 Sections 18 and 19.

1538 For clarity and as required by ZSE, all Devices shall support the Key Establishment Cluster
 1539 as both Client and Server.

1540 A GSME shall implement a ZSE Metering Device and shall implement all the clusters,
 1541 commands, attribute sets and attributes in Table 7.4 where column A is 'GSME: Metering
 1542 Device'.

1543 A GPF shall implement a ZSE Metering Device and shall implement all the clusters,
 1544 commands, attribute sets and attributes in Table 7.4 where column A is 'GPF: Metering
 1545 Device (Gas Mirror Endpoint)'.

1546 A GPF shall implement a ZSE Energy Services Interface and shall implement all the clusters,
 1547 commands, attribute sets and attributes in Table 7.4 where column A is 'GPF: Energy
 1548 Services Interface (Gas ESI Endpoint)'

1549 A CHF shall implement a ZSE Remote Communications Device and shall implement all the
 1550 clusters, commands, attribute sets and attributes in Table 7.4 where column A is 'CHF:
 1551 Remote Communications Device (Remote Communications Endpoint)'.

1552 An ESME which is not a Twin Element ESME shall implement a ZSE Energy Services
 1553 Interface and shall implement all the clusters, commands, attribute sets and attributes in
 1554 Table 7.4 where column A is 'ESME: Energy Services Interface (Electricity ESI Endpoint)'

1555 An ESME which is a Twin Element ESME shall implement three ZSE Energy Services
 1556 Interfaces:

1557 12. the first which shall implement all the clusters, commands, attribute sets and attributes in
 1558 Table 7.4 where column A is 'ESME: Energy Services Interface (Twin ESME aggregate
 1559 ESI Endpoint)';

1560 13. the second which, in relation to the primary measuring element, shall implement all the
 1561 clusters, commands, attribute sets and attributes in Table 7.4 where column A is 'ESME:
 1562 Energy Services Interface (Twin ESME primary/secondary ESI Endpoint)'; and

Table 7.4: Device Requirements

14. the third which, in relation to the secondary measuring
element, shall implement all the *clusters, commands, attribute sets and attributes* in
Table 7.4 where column A is 'ESME: Energy Services Interface (Twin ESME
primary/secondary ESI Endpoint)'.

1567 A PPMID shall implement a ZSE *In-Home Display* and shall implement all the *clusters*,
1568 *commands*, *attribute sets* and *attributes* in Table 7.4 where column A is 'PPMID: In-Home
1569 Display'

1570 An HCALCS shall implement a ZSE Load Control Device and shall implement all the clusters,
1571 commands, attribute sets and attributes in Table 7.4 where column A is 'HCALCS: Load
1572 Control Device'.

1573 An HHT shall implement a ZSE Remote Communications Device and shall implement all the
1574 clusters, commands, attribute sets and attributes in Table 7.4 where column A is 'HHT:
1575 Remote Communications Device'.

1576 An IHD shall support the mandatory attributes of the Basic cluster and the other clusters,
1577 attributes and commands necessary to meet the SMETS requirements.

1578 Where a row in Table 7.4 is required for a Device, that Device shall support the cluster,
1579 attribute or command specified in that row as client or server, as specified in column C
1580 (labelled 'Client / Server').

1581 Support for *clusters*, *commands*, *attribute sets* and *attributes* shall be as defined in columns
1582 B ('Cluster'), D ('Command'), E ('Attribute Set') and F ('Attribute').

1583 Note that the other columns in Table 7.4 are informative and for requirements traceability
1584 only.



GBCS v0.8.1 ZSE
Device Requirements.

8 Encryption of Attributes in Remote Party Messages

In some Use Cases, some attributes are marked as Encrypted.

This Section 8 lays out requirements as to how such Encryption and related Decryption shall be undertaken.

8.1 Approach - informative

Since ZSE and DLMS have differing data types to represent the same attribute of SMETS information, there are some differences in the format of the data that is encrypted. These differences are laid out in this Section 8. However, Encryption and Decryption use the same cryptographic AES GCM primitives in the same way in all cases, regardless of protocol. The usage is the same as that to generate MACs for Remote Party Message protection, and therefore as per the AES GCM approach laid out in the Green Book.

Encryption of SMETS attributes is required when:

- the Supplier reads the amounts held in Time Debt Register [1..2] and Payment Debt Register. Each of these is a single integer value;
- the Supplier reads the values held in the Active Import Register or Secondary Active Import Register. Each of these is a single integer value;
- a Known Party or an Unknown Party reads one or more entries from a Log (with each entry in the specific log having a Log specific structure), specifically:
 - the current or previous Supplier reads the Billing Data Log (excluding the export related parts), the Daily Read Log or the Prepayment Daily Read Log. Note that a previous Supplier is an Unknown Remote Party as far as the meter is concerned;
 - the Supplier, Network Operator, or an Unknown Remote Party reads the Daily Consumption Log or the Profile Data Log (Consumption parts);
- a Device sends an Alert containing a single entry from the Billing Data Log (excluding the export related parts).

8.2 Common requirements

All Encryption shall be Authenticated Encryption which:

- shall use the cryptographic primitives, input value structures and cryptographic material specified in Section 4; and
- shall, for Key Agreement, use the Key Agreement key pair of the Device and the Remote Party which is accessing the data item.

A Device shall, where it stores a data item listed in the Mapping Table as Encrypted, only provide that data in a Remote Party Message in Encrypted form.

Where the Encrypted data item is within a Log, a Command requesting that data shall always have 'from' and 'to' date-times specified.

Where all the octets in the 'from' date-time are 0x00 (excluding the least significant 3 bytes in Blue Book octet string formatted date-times), the Device shall interpret the 'from' field as meaning from the oldest in the Log.

1625 Where all the octets in the 'to' date-time are 0xFF (excluding the least significant 3 bytes in
 1626 Blue Book octet string formatted date-times), the Device shall interpret the 'to' field as
 1627 meaning to the newest in the Log.

1628 Where the Encrypted data item in the Mapping Table is not in a Log, a Command requesting
 1629 that data shall never have 'from' or 'to' date-times specified.

1630 8.3 Key Derivation Inputs

1631 Where a Remote Party Message (1) contains Encrypted data items and (2) contains a
 1632 Supplementary Remote Party ID, then the Encryption Remote Party shall be that identified
 1633 by the Supplementary Remote Party ID. Otherwise, the Encryption Remote Party shall be
 1634 the Remote Party identified in the Grouping Header of the Message.

1635 If the Message is to include a Supplementary Originator Counter generated by the Device
 1636 (see Section 4.3.1.4), then the Encryption Originator Counter shall be the Supplementary
 1637 Originator Counter. Otherwise the Encryption Originator Counter shall be the Originator
 1638 Counter with the value in the Grouping Header of the Message.

1639 In relation to the Key Derivation Function requirements at Section 4.3.3, fields shall be
 1640 populated as follows:

- 1641 • 'value of transaction-id' shall be the concatenation 0x04 || Encryption Originator Counter.
 1642 Note 0x04 ensures this value is not used in any other Key Derivation Function
 1643 invocation save that related to this Encryption / Decryption; and
- 1644 • for Encrypted data items in Responses and Alerts, 'value of recipient-system-title' shall
 1645 be Encryption Remote Party and 'value of originator-system-title' shall be the Device's
 1646 Entity Identifier.

1647 8.4 AAD, Plaintext and Ciphertext

1648 The Plaintext shall be set to the structure and content of the data item(s) as they would have
 1649 been exposed on the Device's HAN interface, if access to them were not constrained to be
 1650 via Encrypted form by this Section 8.4.

1651 AAD shall be set to security control byte (SC) which shall have the value of 0x31 (see
 1652 Section 8.4.1).

1653 The Invocation Counter (IC) shall have a value of 0x00000000.

1654 The Authenticated Encryption MAC (AE MAC) shall be the MAC produced by applying
 1655 Authenticated Encryption to AAD and Plaintext, as defined in *NIST Special Publication 800-
 1656 38D*, with the values specified in this Section 8.4.

1657 Authenticated Encryption (AE) Ciphertext shall be the Ciphertext produced by applying
 1658 Authenticated Encryption to Plaintext, with the values specified in this Section 8.4.

1659 CIPHERED INFORMATION shall be the concatenation:

1660 SC || IC || AE Ciphertext || AE MAC

1661 8.4.1 Meaning of SC - informative

1662 The SC is set to 0x31 to reflect the following:

- 1663 • Bits 3..0 are security suite which is 0b0001 since Security Suite 1 is required;
- 1664 • Bit 4 is set to 0b1 since Authentication of the data is required;
- 1665 • Bit 5 is set to 0b1 since Encryption of the data is required;
- 1666 • Bit 6 is set to 0b0 since Messages containing the encrypted data are unicast; and

- 1667 • Bit 7 is set to 0b0 as per the Green Book.

1668 8.5 Access to sensitive data – COSEM attribute access

1669 Access to sensitive data items shall be via the *Data Protection* class, as specified in the Blue
 1670 Book. The required OBIS codes and associated details for each attribute shall be as
 1671 specified in the ‘SMETS required objects’ tab in the Mapping Table.

1672 The Device shall only allow read access to attributes listed as Encrypted in the Mapping
 1673 Table using the `get_protected_attributes(data)` method of the *Data Protection* class and not
 1674 allow access to any other methods of such objects.

1675 8.5.1 Values of the *Data Protection* class attributes

1676 The values of attributes 1, 3, 4, 5 and 6 of an object of the *Data Protection* class shall be set
 1677 on a Device at manufacture, and those values shall not be capable of amendment except by
 1678 firmware upgrade. For each object of the *Data Protection* class:

- 1679 • `protection_object_list` (attribute 3) shall be a single entry array containing one
 1680 object_definition. Within that single object_definition: class_id, logical_name,
 1681 attribute_index, data_index, restriction_type and restriction_value shall take values as
 1682 per Table 8.5.1;
- 1683 • the value of `protection_parameters_get` (attribute 4) shall be a single entry array
 1684 containing one protection_parameters_element, which shall have the values specified in
 1685 Table 8.5.1;
- 1686 • the value of `protection_parameters_set` (attribute 5) shall be an array containing zero
 1687 entries; and
- 1688 • the value of `required_protection` (attribute 6) shall be 0b01100000 (0x60) where the
 1689 object exposes the `get_protected_attributes(data)` method, since Authenticated
 1690 Encryption is required on the output of the method.

Attribute	Type	Value
<code>protection_type</code>	Enum	(2) authentication and encryption
<code>protection_options</code>	Structure	
<code>transaction_id</code>	octet-string	Empty string
<code>originator_system_title</code>	octet-string	Empty string
<code>recipient_system_title</code>	octet-string	Empty string
<code>other_information</code>	octet-string	Empty string
<code>key_info</code>	Structure	
<code>key_info_type</code> :	Enum	(2) agreed_key
<code>key_info_options</code>	CHOICE	agreed_key_options
<code>agreed_key_info_options</code>	Structure	
<code>key_parameters</code>	octet-string	0x02 (meaning C(0e, 2s ECC CDH))
<code>key_ciphered_data</code>	octet-string	An octet string of length zero

Table 8.5.1: Values of `protection_parameters_element`

1692 8.5.2 Parameters of the `get_protected_attributes` method

1693 The `get_protected_attributes_request` parameter of the `get_protected_attributes` method
 1694 shall:

- 1695 • be populated in the Command to the Device according to Table 8.5.2a; and
 1696 • be verified by the Device receiving the Command according to Table 8.5.2a;
 1697 The protection_parameters part of the get_protected_attributes_response returned by the
 1698 get_protected_attributes method shall be populated by the Device according to Table 8.5.2b.
 1699 The value of protected_attributes part of the protected_attributes_response_data returned by
 1700 the get_protected_attributes method shall be populated by the Device with CIPHERED
 1701 INFORMATION, calculated as per the requirements of Section 8.2. The tag for
 1702 protected_attributes shall be 'octet-string' (0x09) and the length shall be the length of
 1703 CIPHERED INFORMATION.

Field	Value	Device Validation	Note
get_protected_attributes_request			
tag	0x02	Must have this value	Meaning 'structure'
length	0x02	Must have this value	2 elements in the structure
object_list			The first element in the get_protected_attributes_request structure
tag	0x01	Must have this value	Meaning 'array'
length	0x01	Must have this value	1 entry in the array
object_definition			The 1 entry in the object_list array
tag	0x02	Must have this value	Meaning 'structure'
length	0x05	Must have this value	5 elements in the structure
class_id			
tag	0x12	Must have this value	Meaning 'long-unsigned'
value	See 'Note' column	Must be the same as the class_id in attribute 3 of the Data Protection object being accessed	The class_id of the object which is the source of the Encrypted data
logical_name			
tag	0x09	Must have this value	Meaning 'octet-string'
length	0x06	Must have this value	Logical_name is always 6 octets long
value	See 'Note' column	Must be the same as the logical_name in attribute 3 of the Data Protection object being	The logical_name of the object which is the source of the Encrypted data

Field	Value	Device Validation	Note
		accessed	
attribute_index			
tag	0x0F	Must have this value	Meaning 'integer'
value	See 'Note' column	Must be the same as the attribute_index in attribute 3 of the Data Protection object being accessed	The attribute_index of the object which is the source of the Encrypted data
data_index			
tag	0x12	Must have this value	Meaning 'long-unsigned'
value	0x0000	Must have this value	Meaning the whole attribute is captured or set
restriction			
tag	0x02	Must have this value	Meaning 'structure'
length	0x02	Must have this value	2 elements in the structure
EITHER		Must be present if this invocation is not to access a Log as defined in Section 8.2	If this is not to access a Log as defined in Section 8.2
restriction_type			
tag	0x16	Must have this value	Meaning 'enum'
value	0x00	Must have this value	Meaning 'none'
restriction_value			Assumes that the CHOICE does not need encoding since the value of 'restriction_type' defines the CHOICE [Note, there are no tags in the Blue Book for this CHOICE]
tag	0x00	Must have this value	Meaning 'null-data'
OR		Must be present if this invocation is to access a Log as defined in Section 8.2	If this is to access a Log as defined in Section 8.2
restriction_type			
tag	0x16	Must have this value	Meaning 'enum'
value	0x01	Must have this	Meaning 'restriction by date'

Field	Value	Device Validation	Note
		value	
restriction_value			Assumes that the CHOICE does not need encoding since the value of 'restriction_type' defines the CHOICE [Note, there are no tags in the Blue Book for this CHOICE]
tag	0x02	Must have this value	Meaning 'structure'
length	0x02	Must have this value	2 elements in the structure
from_date			In the date-time format of the Blue Book
tag	0x09	Must have this value	Meaning 'octet-string'
length	0x0C	Must have this value	Date-time is always 12 octets long
value	See 'Note' column		Log entries with a date-time stamp prior to this date-time shall not be returned.
to_date			In the date-time format of the Blue Book
tag	0x09	Must have this value	Meaning 'octet-string'
length	0x0C	Must have this value	Date-time is always 12 octets long
value	See 'Note' column		Log entries with a date-time stamp after this date-time shall not be returned.
protection_parameters			The second element in the get_protection_attributes_request structure
tag	0x01	Must have this value	Meaning 'array'
length	0x01	Must have this value	1 entry in the array
protection_parameters_element			The 1 entry in the protection_parameters array
tag	0x02	Must have this value	Meaning 'structure'
length	0x02	Must have this value	2 elements in the structure
protection_type			The first element in the protection_parameters_element
tag	0x16	Must have this value	Meaning 'enum'
value	0x02	Must have this value	Meaning 'authentication and encryption'
protection_options			The second element in the protection_parameters_element
tag	0x02	Must have this value	Meaning 'structure'
Length	0x05	Must have this	5 elements in the structure

Field	Value	Device Validation	Note
		value	
transaction_id			
Tag	0x09	Must have this value	Meaning 'octet-string'
Length	0x09	Must have this value	transaction_id is always 9 octets in length
Value	See 'Note' column		The concatenation 0x04 the Originator Counter value part of the transaction_id in the Grouping Header of this Command
originator_system_title			
Tag	0x09	Must have this value	Meaning 'octet-string'
Length	0x08	Must have this value	Entity Identifier is always 8 octets in length
Value	See 'Note' column		The Entity Identifier of the Encryption Remote Party
recipient_system_title			
Tag	0x09	Must have this value	Meaning 'octet-string'
Length	0x08	Must have this value	Entity Identifier is always 8 octets in length
Value	See 'Note' column	Must be the Device's Entity Identifier	The Entity Identifier of the Device
other_information			
Tag	0x09	Must have this value	Meaning 'octet-string'
Length	0x00	Must have this value	Zero length since this string is empty
key_info			
Tag	0x02	Must have this value	Meaning 'structure'
Length	0x02	Must have this value	2 elements in the structure
key_info_type:			
Tag	0x16	Must have this value	Meaning 'enum'
Value	0x02	Must have this value	Meaning 'agreed_key'
key_info_options		CHOICE	Assumes that the CHOICE does not need encoding since the value of 'restriction_type' defines the CHOICE [Note, there are no tags in the Blue Book for this CHOICE]
agreed_key_info_options			
tag	0x02	Must have this value	Meaning 'structure'

Field	Value	Device Validation	Note
length	0x02	Must have this value	2 elements in the structure
key_parameters			
tag	0x09	Must have this value	Meaning 'octet-string'
length	0x01	Must have this value	Length fixed by the Blue Book
value	0x02	Must have this value	Meaning 'C(0e, 2s ECC CDH)'
key_ciphered_data			
tag	0x09	Must have this value	Meaning 'octet-string'
length	0x00	Must have this value	Zero length since this string is empty

1704

Table 8.5.2a: values of get_protected_attributes_request

Field	Value	Device Validation	Note
protection_parameters			
tag	0x01	Must have this value	Meaning 'array'
length	0x01	Must have this value	1 entry in the array
protection_parameters_element			The 1 entry in the protection_parameters array
tag	0x02	Must have this value	Meaning 'structure'
length	0x02	Must have this value	2 elements in the structure
protection_type			The first element in the protection_parameters_element
tag	0x16	Must have this value	Meaning 'enum'
value	0x02	Must have this value	Meaning 'authentication and encryption'
protection_options			The second element in the protection_parameters_element
tag	0x02	Must have this value	Meaning 'structure'
length	0x05	Must have this value	5 elements in the structure
transaction_id			
tag	0x09	Must have this value	Meaning 'octet-string'
length	0x09	Must have this value	transaction_id in is always 9 octets in length
value	See note		The concatenation 0x04 Encryption Originator Counter
originator_system_title			
tag	0x09	Must have this value	Meaning 'octet-string'
length	0x08	Must have this value	Entity Identifier is always 8 octets in length
value	See		The Entity Identifier of the Device

Field	Value	Device Validation	Note
	note		
recipient_system_title			
tag	0x09	Must have this value	Meaning 'octet-string'
length	0x08	Must have this value	Entity Identifier is always 8 octets in length
value	See note	Must be the Device's Entity Identifier	The Entity Identifier of the Encryption Remote Party
other_information			
tag	0x09	Must have this value	Meaning 'octet-string'
length	0x00	Must have this value	Zero length since this string is empty
key_info		Structure	
tag	0x02	Must have this value	Meaning 'structure'
length	0x02	Must have this value	2 elements in the structure
key_info_type:			
tag	0x16	Must have this value	Meaning 'enum'
value	0x00	Must have this value	Meaning 'agreed_key'
key_info_options		CHOICE	Assumes that the CHOICE does not need encoding since the value of 'restriction_type' defines the CHOICE [Note, there are no tags in the Blue Book for this CHOICE]
agreed_key_info_options			
tag	0x02	Must have this value	Meaning 'structure'
length	0x02	Must have this value	2 elements in the structure
key_parameters			
tag	0x09	Must have this value	Meaning 'octet-string'
length	0x01	Must have this value	Length fixed by the Blue Book
value	0x02	Must have this value	Meaning 'C(0e, 2s ECC CDH)'
key_ciphered_data			
tag	0x09	Must have this value	Meaning 'octet-string'
length	0x00	Must have this value	Zero length since this string is empty

Table 8.5.2b: values of protection_parameters

1705 8.5.3 Billing Data Log Alert – DLMS COSEM

1706 'Use Case Specific Additional Content' within the Billing Data Log Alert shall be populated
 1707 according to the Message Template for the Billing Data Log Alert in Section 18.2.

1708 8.6 Access to sensitive data – ZSE attribute access

1709 Ciphered Information shall be used to populate each 'GBZ Use Case Specific Component
 1710 with encrypted content' as specified in Section 7.2.10.

9 Time Synchronisation and Future Dated Remote Party Messages

This Section 9 details how time synchronisation shall operate, and how future dated Remote Party Messages shall be processed by Devices. The latter applies only where a Command is specified in 'Use Case reference' tab in the Mapping Table, as 'Capable of future dated invocation'.

Note that all references in the GBCS to time shall be to UTC date-time unless explicitly stated otherwise.

9.1 Time synchronisation

9.1.1 Introduction – informative

SMETS requires that ESME and GSME have clocks and that, under normal operating circumstances, the time on those clocks is accurate to within 10 seconds.

CHTS requires that Communications Hubs have clocks and that, under normal operating circumstances, the time on those clocks is accurate to within 10 seconds.

Critical functionality on Communications Hubs can function predictably without reliance on time. Time setting mechanisms on Communications Hubs therefore are not constrained or specified in the GBCS. However, under normal operating circumstances, a Communications Hub will provide the time reference for all dependent Devices on the HAN.

Significant parts of ESME and GSME functionality are time-dependent for their correct and predictable functioning. This includes Critical functionality which can only be controlled by the Device's Supplier with responsibility for that Device. Thus, time must be accurate in terms of alignment with the time set by the Supplier on the ESME / GSME. However, the accuracy requirements measured in seconds are smaller than end-to-end network latency for delivery of Commands to Devices.

This leads to a time synchronisation approach for ESME as specified in Section 9.1.4, and GSME as specified in Section 9.1.6.

That approach is:

- for the Supplier to send a Set Clock Command with the Supplier's current time and a future time (reflecting a time tolerance) in the Command; and
- if, when the Device receives the Command, the Communications Hub's time is within tolerance of the Supplier's time, the Device aligns itself to the Communications Hub's time and treats its time as Reliable. Otherwise the Device treats its time as Unreliable.

The time synchronisation for a GSME follows the same principles but tolerance needs to differ because a GSME is 'sleepy'. 'Sleepy' means that its SMHAN radio will not be active most of the time and therefore the tolerance provided by the Supplier needs to reflect the extended latency.

9.1.2 Common Requirements – Set Clock

Supplier Current Time shall be the Supplier's time at the point the Supplier sends a Set Clock Command.

GSME and ESME shall maintain a record of its Time Status, which, for clarity, is not the same as the ZSE *TimeStatus* attribute in the Remote Communications Endpoint. Time Status shall have one of the values in Table 9.1.2.

1753

Value	Meaning
Invalid	The Device has no meaningful time
Unreliable	The Device has a meaningful time but that time may not be accurate and needs to be affirmed / reaffirmed by the Supplier
Reliable	The Device has a meaningful time and that time has been affirmed by its Supplier

Table 9.1.2: Time Status

9.1.3 Device Requirements relating to the ZCL Time Cluster and its usage

All italicised terms in this Section 9.1.3 shall have the meanings defined in the *Time Cluster* specification within the ZigBee Cluster Library (ZCL) [075123r04ZB].

In relation to the ZCL *Time Cluster* in the Remote Communications Endpoint, a Communications Hub shall:

- set the *Time* attribute to the *UTCTime* provided to it via its WAN interface, whenever such time information is available to it via its WAN interface;
- set the *Time* attribute to 0xFFFFFFFF whenever it cannot accurately maintain its time via its WAN interface to the tolerance required by the CHTS; and
- always have *TimeStatus* attributes set as:
 - Attribute Bit Number 0 (Master) equal to 0b1 (master clock);
 - Attribute Bit Number 2 (MasterZoneDst) equal to 0b0 (not master for Time Zone and DST); and
 - Attribute Bit Number 3 (Superseding) equal to 0b1 (time synchronisation can supersede).

In relation to any ZigBee Time Cluster on the ESME, the ESME shall always have *TimeStatus* attributes set as:

- Attribute Bit Number 0 (Master) equal to 0b0 (not master clock); and
- Attribute Bit Number 3 (Superseding) equal to 0b0 (time synchronisation should not be superseded).

At power on of the clock, an ESME or GSME shall:

- set its Time Status as 'Invalid';
- attempt to synchronise time, using the Communications Hub's Time Cluster; and
- where a valid *Time* (so not 0xFFFFFFFF) is provided by the Communications Hub before any Set Clock Command is received, set its time to the value of *Time* provided and set its Time Status to 'Unreliable'.

ESME and GSME shall attempt to synchronise time, using the Communications Hub's Remote Communications Endpoint Time Cluster, once every 24 hour period in line with the SMETS requirement. ESME and GSME shall undertake the following processing dependent on the outcome of each attempted synchronisation:

- if a time of 0xFFFFFFFF is provided or if no time is received the Device shall retry the synchronisation after an elapsed period of 30 minutes, for a minimum of the lesser of three retries, or a retry resulting in a valid Time (so not 0xFFFFFFFF) being provided. If

- 1788 no valid Time has been provided after three retries, the Device shall set its Time Status
 1789 to ‘Unreliable’;
- 1790 • if the time provided by the Communications Hub differs from the Device’s time by more
 1791 than 10 seconds, then the Device shall:
 1792 o set its Time Status to ‘Unreliable’; and
 1793 o if this results in a change to Time Status, the Device shall construct and issue an
 1794 Alert with Alert Code 0x800C, meaning that its time would have been shifted by
 1795 more than 10 seconds. If the Device is a GSME, the Alert Payload shall be a GBZ
 1796 Alert Payload. If the Device is an ESME, the Alert Payload shall be a DLMS
 1797 COSEM Alert Payload.
- 1798 • if the time provided by the Communications Hub differs from the Device’s time by 10
 1799 seconds or less, then the Device shall adjust its time to the Communications Hub’s time.

9.1.4 ECS70 Set Clock on ESME

1801 This Use Case covers the setting of the Clock by the Supplier on an ESME.

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Command and Response
Message Type Category	SME.C.C
Capable of future dated invocation?	No
Protection Against Replay Required?	Yes
SEC User Gateway Services Schedule (Service Request) Reference	6.11
Valid Target Device(s)	ESME
Valid Business Originator role(s) for Command invocation (and so, for DLMS COSEM Commands, which Application Association is to be used for delivery of the Command to the Device) [Remote Party Messages Only]	Supplier
Valid Response Recipient role(s) (only for Messages authorised by the Access Control Broker on behalf of parties not known to the Device) [Remote Party Messages Only]	N/A
Valid initiating Device type(s) [HAN Only Messages]	N/A
Protocol	DLMS COSEM -

1802 Table 9.1.4: Use Case Cross References for ECS70 Set Clock on ESME

9.1.4.1 Pre-conditions

1803 None.

9.1.4.2 Detailed Steps

1805 The Command Payload shall be constructed as per Table 9.1.4.2a.

Class	OBIS Code	Attribute or Method?	Attribute / Method no.	Set, Get or Action	Attribute/Method name and Blue Book ref.	DLMs COSEM data types	Value (for Sets or Actions)	Notes
8	0-0:1.0.0.255	A	9	Set	clock_base	Enum	5	5 shall mean radio controlled which shall be interpreted as controlled via the Communications Hub Time Cluster that is available over the ESME's HAN interface
8	0-0:1.0.0.255	M	5	Action	preset_adjusting_time	structure{ preset_time: octet-string, validity_interval_start: octet-string, validity_interval_end: octet-string}	{'not specified', Supplier Current Time, Supplier Current Time + Tolerance Period}	'not specified' in preset_time shall be the value 0xFFFFFFFFFFFF80000FF as required by the Blue Book. All times shall be formatted as octet-string according to section 4.1.6.1 of the Blue Book
8	0-0:1.0.0.255	M	4	Action	adjust_to_preset_time	Integer	0	
8	0-0:1.0.0.255	A	2	Get	time	Octet-string	Null	This get means that the resulting Time of the ESME shall be provided in the Response
8	0-0:1.0.0.255	A	4	Get	status	Unsigned	Null	This get means that the resulting Time Status of the ESME shall be provided in the Response

1806 Table 9.1.4.2a: Construction of Command Payload In this Section 9.1.4.2, the object with OBIS
 1807 code 0-0:1.0.0.255 shall be referred to as the *Clock* and italicised terms shall have their Blue
 1808 / Green Book meaning.

- 1809 On receipt of the Command, the ESME shall undertake processing in the following sequence:
- 1810 1. the ESME shall undertake the 'Command Authenticity and Integrity Verification'
 1811 processing required for Commands of type SME.C.C. If that fails, processing shall
 1812 cease;
- 1813 2. the ESME shall process the instructions in the *access-request-body* of the Command as
 1814 follows:
- 1815 a) when attribute 9 of the *Clock* (*clock_base*) is set to '*radio controlled*' (5), the ESME shall
 1816 request the value of the Communications Hub Time from the Communications Hub via
 1817 its ZigBee radio. If a time of 0xFFFFFFFF is provided or if no time is received:
 1818 i. if the current Time Status is set to 'Reliable', the ESME shall set bit 1 of
 1819 attribute 4 (so setting *status* of Clock to be '*doubtful value*' (Unreliable)); or

- 1820 ii. if the current Time Status is not set to 'Reliable', the ESME shall not
 1821 change status.
- 1822 b) the *preset_adjusting_time* method of the Clock shall be executed. Note this is to set
 1823 parameters for the *adjust_to_present_time* method.
- 1824 c) the *adjust_to_present_time* method of the Clock shall be executed as follows:
- 1825 iii. if the Communications Hub Time returned lies between
 1826 *validity_interval_start* and *validity_interval_end*, then:
- 1827 a. ESME *time* shall be updated to match Communications Hub Time;
- 1828 b. the ESME shall unset bit 0 of attribute 4 (so setting *status* of the *Clock*
 1829 not to be an '*invalid value*'); and
- 1830 c. the ESME shall unset bit 1 of attribute 4 (so setting *status* of the *Clock*
 1831 not to be a '*doubtful value*');
- 1832 iv. if the Communications Hub Time returned lies before *validity_interval_start*
 1833 (Supplier Current Time) or after *validity_interval_end* (Supplier Current
 1834 Time + Tolerance)) and (bit 0 of attribute 4 of the *Clock* is unset), then:
- 1835 d. time shall remain unchanged, since time is outside the *validity_interval*;
 1836 and
- 1837 e. the ESME shall set bit 1 of attribute 4 and unset bit 0 of attribute 4 (so
 1838 setting *status* of *Clock* to be a '*doubtful value*' (Unreliable))
- 1839 d) the *get request* on the *time* and *status* attributes of the *Clock* shall be executed;
- 1840 3. the ESME shall undertake the 'Response Construction' and 'Response Cryptographic'
 1841 processing required for a Response of type SME.C.C.

1842 On receipt of the Response, the recipient may undertake the 'Response Recipient
 1843 Processing' for Responses of type SME.C.C.

1844 The meaning of result attributes is as defined in the Green Book.

1845 The meaning of the unsigned integer returned by the get request on attribute 4 of the Clock
 1846 (status) is as per Table 9.1.4.2b.

Values in attribute 4 of the <i>Clock</i> object	Time Status Meaning
Bit 0 is set	Invalid
Bit 0 is unset and Bit 1 is set	Unreliable
Bit 0 is unset and Bit 1 is unset	Reliable

1847 Table 9.1.4.2b: Meaning of unsigned integer

9.1.5 Time related object on ESME

1848 Italicised terms in this Section shall have their Blue Book meaning.

1849 An ESME shall have a *Data* object with OBIS code 0-0:94.44.100.255 where attribute 2 of
 1850 that object:

- 1851 • shall be a double-long-unsigned value;
- 1852 • shall have a value set by the ESME to the number of seconds between 0 hours 0
 1853 minutes 0 seconds on 1st January 2000 UTC and the value of UTC time specified by
 1854 attribute 2 of the *Clock* object with OBIS code 0-0:1.0.0.255;
- 1855 • shall be the value recorded by the ESME in attribute 2 of any *Profile generic* object *entry*
 1856 as the date-time stamp, at the time the *entry* is added;

- 1857 • shall be the format recorded by the ESME in attribute 2 of any *Profile generic* object
 1858 entry in other date-time fields.

1859 Correspondingly:

- 1860 • the ‘from_value’ and ‘to_value’ fields in the *selective access* structure, which are
 1861 required by the GBCS when accessing attribute 2 of any *Profile generic* object directly,
 1862 shall be *double-long-unsigned* attributes containing a date-time specified in seconds
 1863 since 0 hours 0 minutes 0 seconds on 1st January 2000 UTC; and
- 1864 • the *restricting_object* field in the *selective access* structure shall be set with values of
 1865 class_id = 1; logical_name = 0-0:94.44.100.255; attribute_index = 2 and data_index = 0.

1866 The Blue Book requires that, for a *Data Protection* class object, *restriction_by_date* access
 1867 has *from_date* and *to_date* specified as octet-string. Thus, where a Use Case requires that
 1868 the contents of attribute 2 of a *Profile generic* object are returned in Encrypted form (and so
 1869 accessed via a Data Protection object):

- 1870 • the *from_date* and *to_date* fields in the Command shall be octet-strings formatted as per
 1871 section 4.1.6.1 of the Blue Book; and
- 1872 • the ESME shall undertake the conversion necessary to equate these values to
 1873 ‘from_value’ and ‘to_value’ equivalents in accessing attribute 2 of any *Profile generic*
 1874 object.

1875 9.1.6 GCS28 Set Clock on GSME

1876 This Use Case covers the setting of time by the Supplier on a GSME.

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Command and Response
Message Type Category	SME.C.C
Capable of future dated invocation?	No
Protection Against Replay Required?	Yes
SEC User Gateway Services Schedule (Service Request) Reference	6.11
Valid Target Device(s)	GSME
Valid Business Originator role(s) for Command invocation (and so, for DLMS COSEM Commands, which Application Association is to be used for delivery of the Command to the Device) [Remote Party Messages Only]	Supplier
Valid Response Recipient role(s) (only for Messages authorised by the Access Control Broker on behalf of parties not known to the Device) [Remote Party Messages Only]	N/A
Valid initiating Device type(s) [HAN Only Messages]	N/A
Protocol	ASN.1

1877 Table 9.1.6: Use Case Cross References for GCS28 Set Clock on GSME

1878 9.1.6.1 Pre-conditions

1879 None.

1880 **9.1.6.2 Construction of Command**

1881 Set Clock Command Payloads shall be constructed according to the requirements of Section
 1882 9.1.6.4 and populated as specified in Table 9.1.6.2.

1883 MAC Header, Grouping Header, KRP Signature and ACB-SMD MAC shall be populated as
 1884 required for a Command of the SME.C.C Message Category.

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value
@SetTime.CommandPayload	SEQUENCE		
validityIntervalStart	GeneralizedTime	The earliest time the Communications Hub can provide if the Command is to set Reliable Time	Mandatory
validityIntervalEnd	GeneralizedTime	The latest time the Communications Hub can provide if the Command is to set Reliable Time	Mandatory

1885 Table 9.1.6.2: @SetTime.CommandPayload population

1886 **9.1.6.3 Device processing of Command and Response handling**

1887 The GSME receiving a Set Clock Command shall undertake processing steps in the
 1888 sequence defined in this Section 9.1.6.3.

1889 The GSME shall:

- 1890 15. undertake Command Authenticity and Integrity Verification as required for a Command of
 1891 the SME.C.C Message Category;
- 1892 16. request the now current Communications Hub Time. If the Communications Hub cannot
 1893 supply a valid time, it shall provide 0xFFFFFFFF. If this is sent, or no response is
 1894 received, the GSME shall:
 - 1895 a. if its current Time Status is set to 'Reliable', set Time Status to 'Unreliable', set
 1896 deviceTimeStatus to unreliable, populate deviceTime with its current Time
 1897 and process from step 5; or
 - 1898 b. if its current Time Status is not set to 'Reliable', set deviceTimeStatus to be Time
 1899 Status, populate deviceTime with its current Time and process from step 5.
- 1900 17. if ((the Communications Hub Time < validityIntervalStart) or (Communications
 1901 Hub Time > validityIntervalEnd)), set Time Status to 'Unreliable', and set
 1902 deviceTimeStatus to unreliable, leave its Time unchanged, populate
 1903 deviceTime with its current Time and process from step 5;
- 1904 18. set its time to the Communications Hub Time, populate deviceTime with the
 1905 corresponding value, and set deviceTimeStatus to reliable;
- 1906 19. populate the Response Payload according to the requirements of Section 9.1.6.4 using
 1907 the deviceTimeStatus and deviceTime values produced by the processing in this
 1908 Section 9.1.6.3;
- 1909 20. construct MAC Header, Grouping Header and apply the Response Cryptographic
 1910 Protection required for a Response of the SME.C.NC Message Category, and
- 1911 21. send the Response.

1912 On receipt of the Response, the recipient may undertake the 'Response Recipient
 1913 Processing' for Responses of type SME.C.C.

1914 **9.1.6.4 Set Clock Command and Response Payloads - structure definition**

1915 Each instance of @SetTime.CommandPayload and of @SetTime.ResponsePayload
 1916 shall be an octet string containing the DER encoding of the populated structure defined in
 1917 this Section 9.1.6.4 which specifies the structure in ASN.1 notation.

```
1918 SetTime DEFINITIONS ::= BEGIN
1919
1920
1921 CommandPayload ::= SEQUENCE
1922 {
1923   -- specify the period within which the Communications Hub's time must lie
1924   -- if this Command is successfully to set time
1925   validityIntervalStart           GeneralizedTime,
1926   validityIntervalEnd            GeneralizedTime
1927 }
1928
1929 ResponsePayload ::= SEQUENCE
1930 {
1931   -- Specify the Device's now current time
1932   deviceTime                   GeneralizedTime,
1933
1934   -- Specify the Device's now current Time Status
1935   deviceTimeStatus              DeviceTimeStatus
1936 }
1937
1938 DeviceTimeStatus ::= INTEGER
1939 {
1940   reliable                     (0),
1941   invalid                      (1),
1942   unreliable                   (2)
1943 }
1944
1945 END
1946
1947
```

1948 **9.2 Future Dated Remote Party Messages**

1949 **9.2.1 Future Dated Commands for the Reading of Data Items - 1950 informative**

1951 Where future dated execution of a Command to read data items is supported in a Use Case,
 1952 this is achieved by setting values in a schedule stored on the Device. In such cases, the
 1953 sequence of Messages is as follows:

- 1954 • on receipt of a Command to update a schedule, the Device should attempt to
 1955 Authenticate then execute the Command. The Device should then create a
 1956 corresponding Response either indicating the schedule has been set or providing failure
 1957 reasons; and
- 1958 • when each trigger time in the schedule is reached (according to the Clock on the
 1959 Device), the Device undertakes the required processing then creates and sends an Alert.
 1960 One initial Command to set a schedule may generate many such Alerts.

1961 In such circumstances, the Command and Response are specified in one Use Case, and the
 1962 Alert is specified in a different Use Case.

1963 Such future dated reading can be cancelled by sending a Command which resets the
 1964 schedule values.

1965 The only example of such a schedule is the Billing Calendar. The Alerts generated are
1966 Billing Calendar Alerts.

1967 **9.2.2 Future Dated Commands for the Writing of Attributes**

1968 **9.2.2.1 Introduction - informative**

1969 Only Commands marked 'Capable of future dated invocation?' in the Mapping Table can be
1970 future dated. Such Commands allow a data item or group of data items to be changed at a
1971 date-time in the future.

1972 Where a data item or group of data items on a Device are capable of future dated updates,
1973 this is achieved by the Device having:

- 1974 • a 'current' and a 'next' version of the group of data items in question;
- 1975 • a data item for recording the date / time at which the 'next' version should be made
1976 'current'; and
- 1977 • a method to set 'current' values equal to 'next' values.

1978 This is the meaning of data items with 'Current' and 'Next' post fixes in the 'SMETS / CHTS
1979 Attribute / method' column of the Mapping Table.

1980 In such cases, the sequence of Messages to effect a future dated update would be:

- 1981 22. a Command would be sent to the Device instructing that the data items in question
1982 should be stored in the 'next' data items and the activation date / time should be set (so
1983 overwriting previous 'next' values and any previous activation date / time);
- 1984 23. only if the Command is Authenticated, would the Device attempt to execute the
1985 instructions in Command. Execution for such Commands means writing to 'next' values
1986 and setting activation date-times. If the activation date-times are in the past, the Device
1987 will also attempt to make the 'next' values 'current';
- 1988 24. the Device would then create a Response. This Response would either confirm that the
1989 'next' values and the activation date / time have been set (and so any previous future
1990 dated command with this Message Code has been over written), or would provide
1991 failure reasons. The 'current' values would be unaffected assuming the activation date /
1992 time is in the future; and
- 1993 25. when the activation date / time is reached (according to the clock on the Device), the
1994 Device would attempt to make the 'next' values 'current'. The Device would then send
1995 an Alert detailing success or failure.

1996 Like all other Commands, future dated Commands cannot be modified once accepted by the
1997 Device. However, the time activated processing can be stopped from happening by sending
1998 a new Command of the same Message Type. This is because the new Command over
1999 writes the values from the old Command.

2000 For example:

- 2001 • a 'cancellation' can be effected by sending a new Command where the activation date /
2002 time in the new Command has a value that means 'never' to the Device; and
- 2003 • a 'modification' can be effected by sending a new Command where the activation date-
2004 time and / or the 'next' values in the new Command are different than the old one.

2005 Commands that are marked 'Capable of future dated invocation?' in the Mapping Table can
2006 also be invoked immediately, as specified in Section 9.2.2.4.

2007 When there is a change of Supplier on a Device which is (1) after a future dated change is
2008 stored but (2) before it is activated, the processing at Section 13.3.5.10 will be undertaken at

2009 the point of update of Security Credentials. This ensures future dated commands from the
2010 old Supplier will not be actioned by the Device.

2011 **9.2.2.2 Date-times in future datable commands**

2012 Where a Command contains more than one activation date-time field, the values in all
2013 activation date-times in an instance of that Command shall be the same, except for ZSE
2014 Command Payloads where a field contains 0xFFFFFFFF or 0xFFFFFFF, then all activation
2015 date-times shall be either 0xFFFFFFFF or 0xFFFFFFF. Devices shall reject Commands
2016 not complying with this requirement. A future datable Command shall be future dated when
2017 it contains an activation date-time that is not one of the values in Section 9.2.2.4.

2018 **9.2.2.3 Effect on prior Commands of the same Message Code**

2019 On receipt of an Authenticated future datable Command, the Device shall overwrite all parts
2020 of any previously sent future dated Command of the same Message Code and, if the
2021 activation date-times for the instructions in the Command are in the past, the Device shall
2022 execute the instructions immediately.

2023 **9.2.2.4 Using a future dated Command to write Attributes immediately**

2024 Where a Command is marked 'Capable of future dated invocation?' in the Mapping Table,
2025 instructions within the Command shall be executed immediately after Authentication by the
2026 Device when:

- 2027 • for DLMS COSEM Commands Payloads, activation date-time(s) have the value
2028 0x000000000000000000008000FF;
- 2029 • for ZSE Command Payloads, the activation date-time(s) have the value 0x00000000;
2030 and
- 2031 • for ASN.1 Command Payloads, the activation date-time is not present.

2032 **9.2.2.5 Cancellation of future dated Commands for the writing of Attributes**

2033 Where a Command is marked 'Capable of future dated invocation?' in the Mapping Table,
2034 instructions within the Command shall never be executed by the Device when:

- 2035 • for DLMS COSEM Commands Payloads, activation date-time(s) have the value
2036 0xFFFFFFFFFFFFFF8000FF;
- 2037 • for ZSE Command Payloads, the activation date-time(s) have the value 0xFFFFFFFF
2038 for any ZSE command other than PublishCalendar, PublishSpecialDays,
2039 PublishBlockThresholds, PublishPriceMatrix;
- 2040 • for ZSE Command Payloads, the activation date-time(s) have the value 0xFFFFFFF
2041 for the ZSE command PublishCalendar, PublishSpecialDays, PublishBlockThresholds,
2042 PublishPriceMatrix; and
- 2043 • for ASN.1 commands, the activation date-time has a value of 99991231235959Z.

2044 For clarity, sending such a Command has the effect of 'cancelling' any previously sent future
2045 datable Command of the same Message Code that the Device has not already executed.

2046 **9.2.2.6 Reactions to Future Dated Commands**

2047 Subject to Command Authenticity and Integrity Verification as detailed in Section 6, where a
2048 Command is future dated, at time the Command is received, the Device shall send a
2049 Response to the Command:

- 2050 • where activation date-times are in the past or the instructions detail immediate execution
2051 as per Section 9.2.2.4, the Command shall be executed immediately, the Response
2052 shall detail the outcome of the Command's execution, and no Alert shall be generated;

- 2053 • where activation date-times are in the future, that Response shall detail the success or
2054 otherwise of storing the details in the Command; and
- 2055 • if the activation date-times are in the future and the Command's details were
2056 successfully stored, the Device shall, at the time the future activation date-time is
2057 reached, process each of the instructions as specified in the Command in the sequence
2058 specified in that Command and then generate an Alert with an Alert Payload, for each
2059 instruction, of the same type as the Payload type of the Command and an Alert Code of
2060 0x8066 for successful execution and 0x8067 for failed execution. Thus:
- 2061 ○ an ASN.1 Command Payload shall lead to an ASN.1 Alert Payload, which shall be as
2062 defined in Section 13;
- 2063 ○ a DLMS COSEM Command Payload shall lead to DLMS COSEM Alert Payload(s),
2064 which shall be as defined in Table 7.2.9c, where the Use Case specific additional
2065 content contains the concatenation 0x09 || 0x13 || Message Code || Originator
2066 Counter || cosem-attribute-descriptor from the corresponding part of the Command
2067 Payload. Note that 0x09 is the DLMS COSEM tag for octet-string and 0x13 is the
2068 length of the concatenation Message Code || Originator Counter || cosem-attribute-
2069 descriptor; or
- 2070 ○ a GBZ Command Payload shall lead to GBZ Alert Payload(s), shall be as defined in
2071 Table 7.2.10c, where the Use Case specific additional content contains the
2072 concatenation Message Code || Originator Counter || Extended Header Cluster ID
2073 || Frame control || Command identifier from the corresponding part of the Command
2074 payload.

2075 **9.2.2.7 ESME requirements for activation of future datable Commands**

2076 When either (1) a Command successfully sets the 'passive' Account object's
2077 *account_activation_time* (*attribute ID* 13, *OBIS code* 0-1:19.0.0.255) to
2078 0x000000000000000000000000000000008000FF or (2) the ESME's clock reaches the value in that attribute
2079 or (3) the *activate_account* method is invoked, the ESME shall not activate 'passive' Account
2080 object, as detailed in the Blue Book, but shall set the attributes in the 'active' Account object
2081 (*OBIS code* 0-0:19.0.0.255) as follows:

- 2082 • set the *payment_mode* part of the *account_mode_and_status* attribute (*attribute ID* 2) to
2083 the *payment_mode* value in the 'passive' Account object; and
- 2084 • set each of the other *static* attributes (as defined in the Blue Book) to the corresponding
2085 value in the 'passive' Account object.

2086 When either (1) a Command successfully sets an Activation Date-Time Attribute in Table
2087 9.2.2.6 to 0x000000000000000000000000000000008000FF or (2) the ESME's clock reaches the value in an
2088 Activation Date-Time Attribute in Table 9.2.2.6, the ESME shall set the corresponding Active
2089 Attribute in Table 9.2.2.6 to the value of the corresponding Passive Attribute in Table 9.2.2.6.

SMETS Reference	Activation Date-Time Attribute			Passive Attribute			Active Attribute		
	Class ID	OBIS	Attr. ID	Class ID	OBIS	Attr. ID	Class ID	OBIS	Attr. ID
TariffSwitchingTable(SpecialDays)	9000	0-0:94.44.128.29	6	11	0-1:11.0.0.255	2	11	0-0:11.0.0.255	2
StandingCharge.period	9000	0-0:94.44.128.32	6	9000	0-0:94.44.128.32	4	113	0-0:19.2.4.255	8
TariffThresholdMatrix	9000	0-0:63.1.1.255	6	21	0-0:16.1.12.255	2	21	0-0:16.0.12.255	2
TariffSwitching Table(SecondaryElement).specialDays	9000	0-0:94.44.128.30	6	11	0-1:11.0.1.255	2	11	0-0:11.0.1.255	2
DisablementThreshold(MeterBalance)	9000	0-0:94.44.128.22	6	9000	0-0:94.44.128.22	4	21	0-0:16.0.1.255	2
DebtRecoveryRateCap	9000	0-0:94.44.128.12	6	9000	0-0:94.44.128.12	4	111	0-0:19.0.0.255	18
DebtRecoveryRateCap(period)	9000	0-0:94.44.128.13	6	9000	0-0:94.44.128.13	4	111	0-0:19.0.0.255	19
EmergencyCreditLimit	9000	0-0:94.44.128.2	6	9000	0-0:94.44.128.2	4	112	0-0:19.10.1.255	9
EmergencyCreditThreshold	9000	0-0:94.44.128.3	6	9000	0-0:94.44.128.3	4	112	0-0:19.10.1.255	10
LowCreditThreshold	9000	0-0:94.44.128.9	6	9000	0-0:94.44.128.9	4	111	0-0:19.0.0.255	16
Non-DisablementCalendar	9000	0-0:94.44.128.28	6	10	0-0:12.1.1.255	2	10	0-0:12.0.1.255	2
LoadLimitPeriod(Timer)	9000	0-0:94.44.128.6	6	9000	0-0:94.44.128.6	4	71	0-0:17.0.0.255	6
LoadLimitPowerThreshold	9000	0-0:94.44.128.7	6	9000	0-0:94.44.128.7	4	71	0-0:17.0.0.255	4
LoadLimitRestorationPeriod(Timer)	9000	0-0:94.44.128.8	6	9000	0-0:94.44.128.8	4	71	0-0:17.0.0.255	7
AuxiliaryLoadControlSwitchesCalendar	9000	0-0:94.44.128.26	6	10	0-0:12.0.2.255	2	10	0-1:12.0.2.255	2
Non-DisablementCalendar(SpecialDays)	9000	0-0:94.44.128.31	6	11	0-1:11.0.2.255	2	11	0-0:11.0.2.255	2
AuxiliaryLoadControlSwitchesCalendar(SpecialDays)	9000	0-0:94.44.128.35	6	11	0-1:11.0.3.255	2	11	0-0:11.0.3.255	2

2090

Table 9.2.2.7: Values for Active and Passive Attributes for Account objects

10 ZSE Implementation

Italicised terms in this Section 10 shall have their meaning in the ZCL / ZSE specifications.

10.1 Introduction - informative

This Section 10 sets out specific requirements relating to the implementation of ZSE in Devices:

- Tunnels: requirements relating to Devices' support for the *Tunneling Cluster*. This includes specific differences between GSME, HHT and other Devices, related to their use of the *Tunneling Cluster*. Note that all Devices except Type 2 Devices shall support the *Tunneling Cluster*, since this is the mechanism by which Remote Party Messages (and HAN Only Messages between a PPMID and a GSME) are transported over the HAN;
- GSME and GPF interactions (including the Tapping Off Mechanism): this includes requirements relating to the GPF maintaining a copy of GSME data items, where copies are not supported natively by ZSE mirroring;
- GPF structured data items: requirements relating to how structured data items on the GPF are updated by the GSME and resulting values on the GPF are calculated; and
- HHT interactions – requirements relating to HHT connection to the SMHAN, including specific *Tunneling Cluster* related requirements.

10.2 Tunnels

10.2.1 Overview – informative

All Remote Party Messages are carried across the SMHAN using the *Tunneling Cluster's TransferData* command.

Type 2 Devices such as IHDs are not required to send or receive Remote Party Messages and so are not required to support the *Tunneling Cluster*.

Remote Party Messages to and from the GPF do not cross the SMHAN and so do not use the *Tunneling Cluster*.

All other types of Device need to be able to send and receive Remote Party Commands over the SMHAN and so, as specified in Section 10.2.2, shall support the *Tunneling Cluster*.

Section 10.2.2 lays out the associated requirements, across all Devices including those for the GSME and HHT.

GSME requirements are different than all other Devices since a GSME is a 'sleepy' Device. Additional GSME requirements are laid out in Sections 10.2.4 and 10.3.

HHT interactions also have specific requirements due to their function. These specific requirements are laid out in Section 10.5.

A PPMID may be a sleepy device, and therefore may have different requirements to other Type 1 devices.

10.2.2 Requirements for the Tunneling Cluster

Remote Party Messages and SME.C.PPMID-GSME Messages shall be transported over the SMHAN using the *Tunneling Cluster's TransferData* command. Except where a *TransferData* command is to or from a GSME, the value of the *Data* field's payload in the *TransferData* command shall be the Remote Party Message or SME.C.PPMID-GSME

2132 Message. Where a *TransferData* command is to or from a GSME, the *Data* field's payload
2133 of the *TransferData* command shall take the values specified in Section 10.2.4.

2134 Devices supporting the *Tunneling Cluster* as a *Server* shall have a
2135 *MaximumIncomingTransferSize* set to 1500 octets, in line with the ZSE default. All Devices
2136 supporting the *Tunneling Cluster* shall use this value in any *RequestTunnelResponse*
2137 command and any *RequestTunnel* command.

2138 Devices shall set the value of the *ManufacturerCode* field in any *RequestTunnel* command
2139 to 0xFFFF ('not used').

2140 The *ProtocolID* of all Remote Party Messages shall be 6 ('GB-HGRP'). Devices shall set the
2141 value of the *ProtocolID* field in any *RequestTunnel* command to 6.

2142 Devices shall set the value of the *FlowControlSupport* field in any *RequestTunnel* command
2143 to 'False'.

2144 All Devices except Type 2 Devices and GPFs shall support the *Tunneling Cluster* and, within
2145 that Cluster, the use of the protocol with a *ProtocolID* of 6 (GB-HGRP).

2146 An ESME, an HCALCS and a PPMID shall support the *Tunneling Cluster* as a *Server*.

2147 A GSME and an HHT shall support the *Tunneling Cluster* as a *Client*.

2148 A CHF and PPMID shall support the *Tunneling Cluster* as a *Client* and as a *Server*.

2149 A GPF shall support mirroring functionality. The *Basic Cluster Physical Environment*
2150 attribute shall be supported and shall have the value 0x01.

2151 When a Device receives a *CloseTunnel* command, the Device shall not close that tunnel
2152 unless the command is sent from the Device which opened the tunnel.

10.2.2.1 ESME, HCALCS and PPMID

2153 When a Communications Hub has successfully established a shared secret key using *CBKE*
2154 with a Device of type ESME, HCALCS or PPMID, the CHF shall send a *RequestTunnel*
2155 command to the Device to request a tunnel association with the Device.

2156 Where an ESME, a HCALCS or a PPMID remains in the CHF Device Log, the CHF shall
2157 send a *RequestTunnel* command to the Device whenever:

- 2158 • 0xFFFF seconds have elapsed since receipt of the most recent
2159 *RequestTunnelResponse* command from that Device; or
- 2160 • the CHF receives a Remote Party Message addressed to the Device but does not have
2161 a functioning tunnel association with the Device; or
- 2162 • the CHF powers on.

2163 Where the CHF receives a *RequestTunnelResponse* command from a Device with a
2164 *TunnelStatus* of 0x01 (*Busy*), the CHF shall send another *RequestTunnel* command three
2165 minutes later.

2166 Where the CHF receives a *RequestTunnelResponse* command from a Device with a
2167 *TunnelStatus* of 0x02 (*No More Tunnel IDs*), the CHF shall send a *CloseTunnel* command
2168 for any *TunnelID* that may relate to an active tunnel association with that Device and, after
2169 receiving responses to all such commands, send another *RequestTunnel* command.

10.2.2.2 GSME

2171 When a GSME has successfully established a shared secret key using *CBKE* with a
2172 Communications Hub, the GSME shall:

- 2173 • send a request to the *ZigBee Gas ESI Endpoint* requesting the creation of mirrored
2174 *Basic*, *Metering* and *Prepayment Clusters* using the *RequestMirror* command;

- 2176 • configure, using the *ConfigureMirror* command, the *ZigBee Gas Mirror Endpoint* to use
 2177 the two way mirroring notification scheme ‘*Predefined Notification Scheme B*’; and
 2178 • send a *RequestTunnel* command to the CHF to request a tunnel association with the
 2179 CHF.

2180 Where the Communications Hub has successfully actioned a *ConfigureMirror* command, the
 2181 GPF shall set the *Push All Static Data - Basic Cluster*, *Push All Static Data - Metering*
 2182 *Cluster* and *Push All Static Data - Prepayment Cluster* flags.

2183 For clarity, the GSME:

- 2184 • shall not action ZSE / ZCL commands received from the GPF in relation to any of the
 2185 flags within *NotificationFlags2*, *NotificationFlags3* and *NotificationFlags5*;
 2186 • for *NotificationFlags4*, shall only action ZSE / ZCL commands received from the GPF in
 2187 relation to the flags specified in Table 10.2.2.2a.

Bit Number	Waiting Command
6	<i>Get Prepay Snapshot</i>
7	<i>Get Top Up Log</i>
9	<i>Get Debt Repayment Log</i>

2188 Table 10.2.2.2a: flags in *NotificationFlags4* to be actioned by the GSME

- 2189 • for *FunctionalNotificationFlags*, shall only action ZSE / ZCL commands received from
 2190 the GPF in relation to the flags specified in Table 10.2.2.2b:

Bit Number	Waiting Command
0	<i>New OTA Firmware</i>
1	<i>CBKE Update Request</i>
4	<i>Stay Awake Request HAN</i>
5	<i>Stay Awake Request WAN</i>
6-8	<i>Push Historical Metering Data Attribute Set</i>
9-11	<i>Push Historical Prepayment Data Attribute Set</i>
12	<i>Push All Static Data - Basic Cluster</i>
13	<i>Push All Static Data - Metering Cluster</i>
14	<i>Push All Static Data - Prepayment Cluster</i>
15	<i>NetworkKeyActive</i>
21	<i>Tunnel Message Pending</i>
22	<i>GetSnapshot</i>
23	<i>GetSampledData</i>

2191 Table 10.2.2.2b: flags in *FunctionalNotificationFlags* to be actioned by the GSME

- 2192 • shall have access to the *Notification Flags* on the Communications Hub whenever it can
 2193 communicate with the Communications Hub; and
 2194 • shall not provide any metering data to the *ZigBee Gas Mirror Endpoint* until and unless
 2195 the GPF’s Entity Identifier is recorded in the GSME Device Log.

2196 The GSME shall send a *RequestTunnel* command to the CHF to request a tunnel
 2197 association with the CHF whenever it does not have a currently valid tunnel association with
 2198 the CHF, and one of the following is true:

- 2199 • the GSME has created an Alert or Response that is to be sent; or

- 2200 • the GSME has ascertained, via the *Tunnel Message Pending* flag, that there is a
2201 Command for it buffered on the Communications Hub.

2202 Where the GSME receives a *RequestTunnel/Response* command from the CHF with a
2203 *TunnelStatus* of 0x01 (*Busy*), the GSME shall send another *RequestTunnel* command the
2204 next time it turns its HAN Interface on.

2205 Where the GSME receives a *RequestTunnel/Response* command from the CHF with a
2206 *TunnelStatus* of 0x02 (*No More Tunnel IDs*), the GSME shall send a *CloseTunnel* command
2207 for any *TunnelID* that may relate to an active tunnel association between it and the CHF and,
2208 after receiving responses to all such commands, send another *RequestTunnel* command.

2209 **10.2.3 GSME Tunnel Management – informative**

2210 Commands are sent from the Communications Hub via the tunnel to the GSME. Since the
2211 GSME is a ‘sleepy’ Device, a mechanism is needed for the GSME to request that
2212 Commands are sent to it by the CHF.

2213 In common with the transport of all Remote Party Messages, the mechanism used is the
2214 *TransferData* command, but *TransferData* commands sent between a GSME and CHF need
2215 to distinguish between when:

- 2216 • the GSME is sending a Remote Party Message, so an Alert or a Response or a GBT
2217 Message containing part of an Alert / Response;
- 2218 • the GSME is asking the CHF to send it a Command, or a GBT Message containing part
2219 of a Command; and
- 2220 • the CHF is sending the GSME a Command, or a GBT Message containing part of a
2221 Command.

2222 To meet this need, the following sections specify additional structure in the first part of the
2223 *Data* parameter of the *TransferData* commands sent between GSME and CHF. Specifically
2224 the sending Device shall:

- 2225 • where a Remote Party Message is being sent, set the *Data* parameter payload in a
2226 *TransferData* command to the concatenation:
2227 Tunnel Manager Header || Remote Party Message
- 2228 • where a Remote Party Message is not being sent (so when the GSME is requesting that
2229 a Message is sent), set the *Data* parameter payload in a *TransferData* command to the
2230 value of Tunnel Manager Header.

2231 A mechanism is also required to notify the GSME that one or more Commands are available
2232 for retrieval from the CHF.

2233 The ZSE specification has a flag called *Tunnel Message Pending* in the *Functional Flag*
2234 *Notification* definition. This flag is used to notify a GSME that the CHF has a Remote Party
2235 Message waiting to be transferred to the GSME. The flag is set on the first pending
2236 Command and is reset when all Remote Party Messages have been transferred to the
2237 GSME. The flag is available through the *ReadAttribute* or *MirrorAttributeResponse*
2238 command. The requirements for setting this flag are specified in Section 10.3.4. The Tunnel
2239 Manager Header identifies three different kinds of *TransferData* command usage:

- 2240 • GET (the value 0x01): this is used by the GSME to retrieve waiting Message from the
2241 CHF;
- 2242 • GET-RESPONSE (the concatenation 0x02 || number of Remote Party Messages
2243 remaining): this is used by the CHF to send a Remote Party Message to the GSME. It
2244 also indicates how many Remote Party Messages have yet to be retrieved; and

- 2245 • PUT(the value 0x03): this is used by the GSME to send a Message via the CHF.
2246 Where a Command is waiting on the CHF for the GSME to retrieve it, the following sequence
2247 shall apply:
2248 26. the *Tunnel Message Pending* flag is set on the Communications Hub as detailed in
2249 Section 10.3.4;
2250 27. the GSME turns on its HAN Interface and obtains the value of the *Tunnel Message*
2251 *Pending* flag; and
2252 28. If the *Tunnel Message Pending* flag is set:
2253 e) the GSME sends a *TransferData* command to the CHF with the GET structure in the
2254 Tunnel Manager Header. The Tunnel Manager Header is the only content in the *Data*
2255 field of this *TransferData* command;
2256 f) the CHF sends a *TransferData* command to the GSME with the GET-RESPONSE
2257 structure in the Tunnel Manager Header and a Message in the remaining part of the
2258 *Data* field of the command. The GET-RESPONSE structure details how many more
2259 Messages are available for retrieval; and
2260 g) the GET and GET-RESPONSE pattern repeats until all Messages have been
2261 transferred or the GSME decides to stop requesting Messages.

2262 When the GSME wishes to send a Message, the GSME sends a *TransferData* command to
2263 the CHF with the PUT structure in the Tunnel Manager Header and the Message in the
2264 remainder of the *Data* field in the *TransferData* command.

2265 **10.2.4 TransferData commands sent between GSME and CHF**

2266 When it wishes to send a Message, so an Alert or Response or GBT Message, a GSME
2267 shall send a *TransferData* command to the CHF with the value in the *Data* parameter
2268 payload of the *TransferData* command set to the concatenation:

2269 0x03 || Message

2270 When it wishes to retrieve a Message stored for it on a CHF, a GSME shall send a
2271 *TransferData* command to the CHF with the value in the *Data* field set to 0x01. When the
2272 CHF receives such a *TransferData* command from a GSME, the CHF shall send a
2273 *TransferData* command to the GSME with the value in the *Data* parameter payload set to:

- 2274 • the concatenation
2275 0x02 || (Number of Messages remaining for retrieval after this Message) || (Message
2276 addressed to the GSME)
2277
2278 where it has Messages for the GSME not yet downloaded by the GSME; or
2279 • the concatenation 0x02 || 0x00, where it has no Messages for the GSME to retrieve, the
2280 0x00 representing the number of Messages available to retrieve.

2281 **10.3 GSME and GPF interactions**

2282 **10.3.1 Introduction - informative**

2283 The GSME is informed that Remote Party Commands are available for it to retrieve via
2284 *Tunnel Message Pending* flag on the GPF.

2285 The GSME should, under normal operating circumstances, retrieve all Commands buffered
2286 for it when it turns its HAN Interface on. For example, if two Commands are buffered for it,
2287 the GSME should retrieve both Commands before turning its HAN Interface off.

2288 However, in some circumstances, a GSME may choose not to retrieve all buffered
2289 Commands in a single session. In such cases, the GSME should retrieve each Command
2290 as soon as possible after that Command is received by the CHF.

2291 Potential reasons for a GSME failing to retrieve all buffered Commands include:

- 2292 • the GSME battery requires time to recover;
- 2293 • the GSME is entering a 'low battery' mode and limiting the use of its radio; or
- 2294 • a radio communications error.

2295 Section 10.3 details actions the CHF may take where Commands, or GBT Messages
2296 containing parts of Commands, for a GSME are not retrieved by the GSME.

2297 Commands addressed to a GSME must be processed by the GSME and, when successfully
2298 processed, any changed operational or configuration data must be made available to the
2299 GPF. The GPF then has updated information to provide to other Devices on the same
2300 SMHAN.

2301 In ZSE terms, the GPF incorporates two distinct logical Devices, which are discoverable and
2302 addressed on different *endpoints*. Section 7 describes which *clusters* reside on which
2303 *endpoint*.

2304 **10.3.2 GSME data residing on the ZigBee Gas Mirror Endpoint - 2305 informative**

2306 The *ZigBee Gas Mirror Endpoint* provides a 'reflection' of the data held by the GSME. A
2307 GSME is typically a battery-powered Device and its HAN Interface is mostly not turned on,
2308 making it unable to respond to other Devices. The GSME turns its HAN Interface on at
2309 regular intervals (e.g. 30 minutes) and pushes consumption data to the *ZigBee Gas Mirror
2310 Endpoint*. This provides other Devices on the same SMHAN with access to GSME
2311 consumption data at any time.

2312 **10.3.3 GSME data residing on the ZigBee Gas ESI Endpoint - 2313 informative**

2314 The *ZigBee Gas ESI Endpoint* holds GSME data which is provided by a Remote Party, for
2315 example pricing. The *ZigBee Gas ESI Endpoint* makes this type of data available to Devices
2316 on the same SMHAN.

2317 GSME data from a Remote Party is sent to the GSME in a Remote Party Command. Such a
2318 Command has to be validated by the GSME before any data in it is applied by the GSME.
2319 For example, a Command to change tariff must be rejected by the GSME if it fails
2320 authentication, and the data in the Command must not be applied in such circumstances.

2321 If data in a Remote Party Command is accepted by the GSME, a mechanism is needed to
2322 provide the changed data to the *ZigBee Gas ESI Endpoint*. This is so that the *ZigBee Gas
2323 ESI Endpoint* can then provide that data to other Devices on the same SMHAN.

2324 A mechanism is also needed to deal with a Response not being received from the GSME.
2325 The lack of a Response may indicate that the GSME and the *ZigBee Gas ESI Endpoint* do
2326 not contain the same value in one or more data items. If data items on the two are not
2327 synchronised, Devices on the SMHAN will display incorrect information.

2328 There are several possible reasons why this lack of a Response may arise, not all of which
2329 mean that data is out of synchronisation:

- 2330 • the Command has failed validation by the GSME and has been discarded;
- 2331 • the Response has been lost due to a communications error; or

- 2332 • a software error.

10.3.4 GSME Command retrieval and TOM Requirements

10.3.4.1 TOM Commands and Responses

2335 A Command shall be a TOM Command if it is a Remote Party Command with one of the
2336 following Message Codes:

- 2337 • 0x006B (GCS01a Set Tariff and Price on GSME);
2338 • 0x006F (GCS05 Update Prepayment Configurations on GSME) – the GPF shall only
2339 process the *Calendar* cluster ZSE commands within the Command;
2340 • 0x0071 (GCS07 Send Message to GSME);
2341 • 0x0015 (CS11 Clear ZigBee Device Event Log) where the Command is addressed to
2342 the GSME;
2343 • 0x007C (GCS23 Set CV and Conversion Factor Value(s) on the GSME);
2344 • 0x007E (GCS25 Set Billing Calendar on the GSME);
2345 • 0x0088 (GCS44 Write Contact Details on GSME); or
2346 • 0x00A3 (GCS01b Set Price on GSME).

2347 A TOM Response shall be a Response to a TOM Command.

2348 For clarity, neither a TOM Response nor a TOM Command may contain Encrypted data.

10.3.4.2 Processing of Commands addressed to a GSME

2350 The CHF, GPF and GSME shall undertake the processing steps below following receipt of a
2351 Remote Party Command by the Communications Hub, where that Command is addressed to
2352 a GSME on the same SMHAN:

- 2353 29. the CHF shall buffer the Command and instruct the GPF to set the *Tunnel Message Pending*
2354 flag to inform the GSME that the Command is awaiting retrieval. If the
2355 Command has been sent as multiple GBT Messages, the GPF *Tunnel Message Pending*
2356 flag shall only be set once all GBT Messages making up the Command have
2357 been received by the Communications Hub. If not all GBT Messages making up a
2358 Command have been received by a Communications Hub within 24 hours of the first
2359 GBT Message in that Command being received, then the CHF may discard the GBT
2360 Messages that have been received for that command;
- 2361 30. if 24 hours elapse after setting the GPF *Tunnel Message Pending* flag without the
2362 Command being retrieved by the GSME, the CHF may discard the Command. If the
2363 CHF discards a Command in this way, it shall notify the GPF and the GPF shall log the
2364 event in its Event Log and send an Alert with a GBZ Payload containing an Alert Code
2365 0x809D;
- 2366 31. when the GSME turns its HAN Interface on, it shall read the *Tunnel Message Pending*
2367 flag and retrieve the Command using the *TransferData* command as defined in Section
2368 10.2.3. Each *TransferData* command received by the GSME shall result in the GSME
2369 sending a *DefaultResponse* command;
- 2370 32. the CHF shall process the *DefaultResponse* commands it receives to establish when the
2371 Command has successfully been retrieved by the GSME, and shall provide an indication
2372 to the GPF accordingly. The GPF shall, when there are no further Commands or GBT
2373 Messages pending retrieval by the GSME, clear the *Tunnel Message Pending* flag;
- 2374 33. if a Command is a TOM Command, the CHF shall retain a copy of the Command
2375 contents. For each such Command, the CHF shall start a response timer at the point

- 2376 where it has received *DefaultResponse* command(s) confirming the GSME has
2377 successfully retrieved the Command;
- 2378 34. once a Command is successfully retrieved by the GSME, the GSME shall process the
2379 Command in line with the requirements of the GBCS. Note that (1) this processing shall
2380 result in the GSME attempting to send a Response to the Command or an Alert that it
2381 has received an invalid Command and (2) if sending a Response, the Response shall,
2382 as per the GBCS requirements, detail the success or failure of GSME processing for
2383 each instruction within the corresponding Command;
- 2384 35. the GSME shall not, under normal operating conditions, delay sending the Response
2385 and shall, where possible, send it before turning its HAN Interface off;
- 2386 36. on receipt of a Response that is a TOM Response, the CHF shall inspect the Response
2387 from the GSME. If the Response indicates successful execution of at least one
2388 elemental ZCL / ZSE command in the corresponding TOM Command, the CHF shall
2389 transfer a copy of the corresponding TOM Command contents and the TOM Response
2390 to the GPF, and shall clear the response timer for the Command;
- 2391 37. on receipt of a TOM Response and the corresponding TOM Command contents, the
2392 GPF shall clear any stored copy it has of a TOM Command and then:
- 2393 ○ if the TOM Command is not future dated, process the elemental ZCL / ZSE
2394 commands contained within the Command according to the *status* within the
2395 Response, updating data it holds accordingly. Once processed by the GPF, the
2396 GPF shall make any updated data available over the WAN and over the HAN to the
2397 Devices in the GPF's Device Log;
- 2398 ○ if the TOM Command is future dated, store a copy of the TOM Command without
2399 updating any data it makes available over the WAN or HAN;
- 2400 38. if a Response to a TOM Command has not been received by the Communications Hub
2401 when the corresponding response timer reaches 6 hours:
- 2402 ○ the CHF may discard its copy of the TOM Command contents, clear the response
2403 timer and notify the GPF accordingly; and
- 2404 ○ on receipt of such a notification, the GPF shall log the event in its Event Log and
2405 send an Alert with a GBZ Payload containing an Alert Code 0x809E;
- 2406 39. for clarity, the CHF shall relay all Remote Party Responses received on its HAN
2407 Interface through the WAN interface;
- 2408 40. whenever the CHF receives an Alert detailing activation of a future dated ZCL / ZSE
2409 command from within a TOM Command (so an Alert with Alert Code 0x8066 where the
2410 Message Code in the Alert Payload is 0x006B or 0x00A3), the CHF shall pass a copy of
2411 that Alert to the GPF;
- 2412 41. on receipt of such an Alert the GPF shall compare the Originator Counter in the Alert
2413 Payload with the Originator Counter of any copy of a TOM Command it holds with the
2414 same Message Code as in the Alert Payload, and:
- 2415 ○ if the Originator Counters match, the GPF shall update the data it shares over the
2416 HAN and WAN with the elemental ZCL / ZSE command contained within the TOM
2417 Command; or
- 2418 ○ if the Originator Counters do not match or the GPF does not hold a TOM Command
2419 with this Message Code, the GPF shall send an Alert with Alert Code 0x809E, as
2420 specified by Section 16.

10.4 GPF Structured Data Items

Underlined terms in this Section 10.4 shall have their meaning in the SMETS and / or CHTS.

10.4.1 Introduction – informative

There are GPF requirements to store structured data items which do not have a direct one to one mapping in ZSE, or the interpretation may be uncertain. These structured data items have to be constructed by the GPF.

10.4.2 Structured Data Items

This Section 10.4.2 details how each structured data item shall be constructed by the GPF.

10.4.2.1 Daily Read Log

The GSME shall record the Daily Read Log data items at midnight UTC as defined in SMETS. In ZSE terms, the GSME shall take a *snapshot* of the relevant items. Note that the format and data of the *snapshot* taken is dependent upon the operating tariff. For example if the GSME tariff is ‘TOU only’, the snapshot shall not capture the block values.

The GSME shall use the *snapshot cause* ‘General’ (0x0001) for the *snapshot* taken.

The GSME shall push the *snapshot* to the GPF using the *PublishSnapshot* command. It is not necessary for the GSME to report any attributes which duplicate those contained in the *snapshot*.

The GPF shall populate the relevant attributes upon receipt of the *PublishSnapshot* command, providing the command is received between midnight (UTC) and the next scheduled wake of the GSME.

The GPF shall store the data contained in the *PublishSnapshot* command in the GPF copy of the GSME Daily Read Log.

In the event of a communications outage, the GPF shall retrieve missing *snapshots* using the *GetSnapshot* command, with the UTC start time field populated based on the last received *snapshot* timestamp, if one has been received.

10.4.2.2 Prepayment Daily Read Log

If the GSME is operating in prepayment mode it shall record the Prepayment Daily Read Log data items at midnight UTC. In ZSE terms, the GSME shall take a *prepayment snapshot* of the relevant items. The format and data of the *prepayment snapshot* taken is defined in ZSE.

The GSME shall use the *snapshot cause* ‘General’ (0x0001) for the *prepayment snapshot* taken.

The GSME shall push the *prepayment snapshot* to the GPF using the *Publish Prepay Snapshot* command.

The GPF shall populate the relevant attributes upon receipt of the *Publish Prepay Snapshot* command, providing the command is received between midnight (UTC) and the next scheduled wake of the GSME.

The GPF shall store the data contained in the *Publish Prepay Snapshot* command in the GPF copy of the GSME Prepayment Daily Read Log.

In the event of a communications outage, the GPF shall retrieve missing *prepayment snapshots* using the *GetPrepaySnapshot* command (and *GetPrepaySnapshot* notification flag) with the UTC start time field populated based on the last received *prepayment snapshot* timestamp, if one has been received.

2463 **10.4.2.3 Billing Data Log - informative**

2464 SMETS defines Billing Data Log as ‘a log capable of storing the following UTC date and time
2465 stamped entries:

- 2466 • twelve entries comprising Tariff TOU Register Matrix, the Consumption Register and
2467 Tariff Block Counter Matrix;
- 2468 • five entries comprising the value of prepayment credits;
- 2469 • ten entries comprising the value of payment-based debt payments; and
- 2470 • twelve entries comprising Meter Balance, Emergency Credit Balance, Accumulated
2471 Debt Register, Payment Debt Register and Time Debt Registers [1 … 2].

2472 Requirements for each part are detailed separately in the following sections.

2473 **10.4.2.4 Billing Data Log - Tariff TOU Register Matrix, the Consumption Register and 2474 Tariff Block Counter Matrix**

2475 The GSME shall capture this *snapshot* at the following trigger points:

- 2476 • End of Billing Cycle (snapshot cause “End of Billing Period”);
- 2477 • Change of Payment Mode (snapshot cause “Change of Meter Mode”);
- 2478 • Change of Tariff (snapshot cause ‘Change of Tariff Information’); and
- 2479 • as specified in Section 13.3.5.10 (snapshot cause ‘Change of Supplier’).

2480 When it next turns on its HAN Interface, the GSME shall push this *snapshot* to the GPF
2481 using the *PublishSnapshot* Command.

2482 The GPF shall store the data contained in the *PublishSnapshot* command in the GPF copy
2483 of the GSME Billing data Log.

2484 In the event of a communications outage, the GPF shall retrieve missing *snapshots* using
2485 the *GetSnapshot* command (and the relevant notification flag) with the UTC start time field
2486 populated based on the last received *snapshot* timestamp, if one has been received, or
2487 0x0000 otherwise.

2488 **10.4.2.5 Billing Data Log - value of prepayment credits**

2489 Upon completion of processing of a valid prepayment top-up, the GSME shall push the latest
2490 five prepayment top-ups to the GPF using the *PublishTop Up Log* command.

2491 The GPF shall store the data contained in the *Publish Top Up Log* command in the GPF
2492 copy of the GSME Billing data Log.

2493 If there has been a communications outage, the GPF shall use the *Get Top Up Log*
2494 command to retrieve all prepayment top-ups that may have been processed during the
2495 communications outage. The GSME shall set the *Date / Time* field of the *Get Top Up Log*
2496 command to the current UTC time.

2497 **10.4.2.6 Billing Data Log - payment-based debt payments**

2498 Upon completion of processing of a valid prepayment top-up where the GSME has made a
2499 debt payment using part of that top-up, the GSME shall push details of that debt payment
2500 only to the GPF using the *Publish Debt Log* command.

2501 The GPF shall record the details provided in the GPF copy of the GSME Billing Data Log.

2502 In cases of communications outages, the GPF shall request any outstanding payment-based
2503 debt payments by use of the *GetDebtRepaymentLog* command (and
2504 *GetDebtRepaymentLog* notification flag) with the Debt Type field set to 0x02 (Debt 3).

2505 **10.4.2.7 Billing Data Log - Meter Balance, Emergency Credit Balance, Accumulated**
2506 **Debt Register, Payment Debt Register and Time Debt Registers [1 ... 2]**

2507 The GSME shall capture this snapshot at the following trigger points:

- 2508 • End of Billing Cycle (snapshot cause bit 1 set: 'End of Billing Period' , as per
2509 PublishSnapshot command);
- 2510 • Change of Payment Mode (snapshot cause bit 14 set: 'Change of Meter Mode');
- 2511 • Change of Tariff (snapshot cause at least one of the bits set: bit 3 'Change of Tariff
2512 Information' and / or bit 4 'Change of Price Matrix' and / or bit 5 'Change of Block
2513 Thresholds'); and
- 2514 • as specified in Section 13.3.5.10 (snapshot cause 'Change of Supplier').

2515 When it next turns on its HAN Interface, the GSME shall push this *snapshot* to the GPF
2516 using the *Publish Prepay Snapshot* command.

2517 The GPF shall store the data contained in the *Publish Prepay Snapshot* command in the
2518 GPF copy of the GSME Billing Data Log.

2519 In the event of a communications outage, the GPF shall retrieve missing *snapshots* using
2520 the *GetPrepaySnapshot* command (and *GetPrepaySnapshot* notification flag) with the UTC
2521 start time field populated based on the last received snapshot timestamp, if one has been
2522 received.

2523 **10.4.2.8 GPF Profile Data Log**

2524 The GPF shall create the GPF Profile Data Log from the consumption information pushed by
2525 the GSME each half hour.

2526 The GSME shall, on each half hour, record the following information and push to the GPF:

- 2527 • the *CurrentSummationDelivered* attribute containing total consumption value (with units
2528 of m³);
- 2529 • the *CurrentDayAlternative ConsumptionDelivered* attribute containing total consumption
2530 today (with units of kWh); and
- 2531 • the *CurrentDayCostConsumptionDelivered* attribute containing total cost of consumption
2532 today (with units of Currency Unit);

2533 Upon receipt of the pushed data, the GPF shall calculate the consumption with units of m³
2534 over the previous half hour by subtracting its previously recorded total consumption value
2535 from the total consumption value now sent.

2536 The resulting value shall be stored in the GPF Profile Data Log.

2537 In the event that there are missing values in the GPF Profile Data Log, the GPF shall
2538 interrogate the GSME Profile Data Log using the *GetSampledData* (*SampleID 0x0000*)
2539 command and the *GetSampledData* notification flag to retrieve missing values.

2540 **10.4.2.9 GPF Daily Gas Consumption Log**

2541 The GPF shall create the GPF Daily Gas Consumption Log based on the values pushed
2542 from the GSME. The difference between last total consumption value pushed from the
2543 GSME each UTC day and the last value pushed in the prior UTC day shall be time stamped
2544 and stored in the GPF Daily Gas Consumption Log, so that the values in the log represent
2545 consumption in that UTC day.

2546 In the event of communications outages resulting in the final daily value being missed, the
2547 GPF shall retrieve the values from the GSME Profile Data Log using the *GetSampledData*
2548 (*SampleID 0x0000*) command and *GetSampledData* notification flag.

10.4.2.10 Historical Attributes

2549 A GSME shall support:

- 2551 • the *Historical Cost Consumption Information* attribute set, measured in Currency Units;
2552 and

- 2553 • the *Alternative Historical Consumption* attribute set, measured in kWh.

2554 A GPF shall mirror the attribute sets listed above.

2555 As per Section 10.4.2.8, the GSME shall, on each half hour, record the following information
2556 and push to the GPF:

- 2557 • total consumption value (with units of m³);
2558 • total consumption today (with units of kWh); and
2559 • total cost of consumption today (with units of Currency Unit);

2560 Using the 'total consumption today' value, the GPF shall update the attributes of the mirrored
2561 *Alternative Historical Consumption* attribute set.

2562 Using the 'total cost of consumption today' value, the GPF shall update the attributes of the
2563 mirrored *Historical Cost Consumption Information* attribute set.

2564 In exception circumstances, the GPF shall request the GSME to push the historical data sets
2565 using the '*Push Historical Metering Data Attribute Set*' and '*Push Historical Prepayment Data*
2566 *Attribute Set*' notification flags. The GSME shall interpret the '*Push Historical Metering Data*
2567 *Attribute Set*' notification flag as requiring it to push the *Alternative Historical Consumption*
2568 attribute set.

10.4.2.11 Other attributes

2570 The GSME shall populate the *AccumulatedDebt* attribute in line with the SMETS
2571 Accumulated Debt requirements, and all other Devices shall interpret that attribute
2572 correspondingly.

2573 The GSME shall populate the *Credit Remaining* attribute in line with the SMETS Meter
2574 Balance requirements, and all other Devices shall interpret that attribute correspondingly.
2575 The GSME shall apply functionality related to the *CutOffValue* attribute in line with this
2576 interpretation of Credit Remaining and the SMETS requirements for Disablement Threshold.

2577 The GPF shall calculate the value of the price in any ZCL *PublishPrice* command it creates
2578 using the tariff information it has derived through the TOM and the time from the
2579 Communications Hub's clock.

2580 The ESME shall populate the *AuxSwitchNLabel* attribute in line with the SMETS Auxiliary
2581 Load Control Switch [n] Description requirements, and all other Devices shall interpret that
2582 attribute correspondingly.

2583 The *CommodityType* and *MeteringDeviceType* attributes shall be set by devices as follows:

- 2584 • 'GPF: Metering Device (Gas Mirror Endpoint)': 128 (Mirrored Gas Metering);
- 2585 • 'GSME: Metering Device': 1 (Gas Metering);
- 2586 • 'ESME: Energy Services Interface (Electricity ESI Endpoint)' and not a polyphase ESME:
2587 0 (Electric Metering);
- 2588 • 'ESME: Energy Services Interface (Electricity ESI Endpoint)' and a polyphase ESME: 15
2589 (Electric Metering Element/Phase 3);
- 2590 • 'ESME: Energy Services Interface (Twin ESME aggregate ESI Endpoint)': 0 (Electric
2591 Metering);

- 2592 • ‘ESME: Energy Services Interface (Twin ESME primary ESI Endpoint)’:13 (Electric
2593 Metering Element/Phase 1); and
2594 • ‘ESME: Energy Services Interface (Twin ESME secondary ESI Endpoint)’:14 (Electric
2595 Metering Element/Phase 2).

2596 When processing a ZSE *Get Event Log* command or a ZSE *Clear Event Log* command with
2597 a Log ID nibble of 0x6 (GSME Event Log) or 0x7 (GSME Security Log), a GPF shall process
2598 the command using the relevant GSME Proxy Log copy of the GSME Event or Security Log.
2599 Other values of Log ID shall refer to the GPF’s own logs.

2600 Where an ESME is not a twin element ESME it shall populate the SiteID attribute with the 13
2601 most significant octets being the Import MPAN and the following 13 octets the Export MPAN.

2602 Where an ESME is a twin element ESME it shall populate:

- 2603 • the SiteID attribute in the ‘ESME: Energy Services Interface (Twin ESME aggregate ESI
2604 Endpoint)’ with the 13 most significant octets being the Import MPAN on the primary
2605 element and the following 13 octets the Export MPAN; and
2606 • the SiteID attribute in the ‘ESME: Energy Services Interface (Twin ESME secondary ESI
2607 Endpoint)’ with the most significant 13 octets being the Import MPAN on the secondary
2608 element.

2609 Where a ZCL / ZSE command containing *IssuerEventId* and / or *ProviderID* fields is received
2610 by a Device as part of a GBZ Remote Party Command, the Device shall undertake no
2611 processing in relation to those two fields. For clarity, this means the Device shall not use the
2612 contents of those fields for anti replay purposes.

2613 ESME shall support *StartRandomizedMinutes* (identifier 0x0000) and
2614 *EndRandomizedMinutes* (identifier 0x0000) attributes on the *Demand Response and Load
2615 Control Cluster* as a Server.

2616 In ZSE *GetSampledData* and *GetSampledDataResponse* commands:

- 2617 • the *SampleID* field shall be interpreted as:
2618 ○ 0x0000 meaning Profile Data Log;
2619 ○ 0x0001 meaning Daily Consumption Log; and
2620 ○ 0x0002 meaning Network Data Log; and
2621 • the *SampleRequestInterval* field shall contain 0xFFFF whenever the *SampleID* field is
2622 0x0001.

2623 A GSME shall reject any *PublishPriceMatrix* command that does not contain four *Price* fields.

2624 When processing a Get Snapshot or Get Prepay Snapshot command, a Device shall return
2625 all snapshots where the Snapshot Cause in the snapshot matches any of the set bits in the
2626 Snapshot Cause parameter of the command. When processing a command where Issuer
2627 Calendar ID has the value 0xFFFFFFFF or 0xFFFFFFF, a GPF or GSME shall interpret
2628 0xFFFFFFFF as meaning the currently in force Tariff Switching Table calendar and
2629 0xFFFFFFF as meaning the currently in force Non-Disablement Calendar.

2630 Devices shall support the requirements relevant to their device type detailed in ‘Trust Center
2631 swap-out process’ section of the ZSE specification except that:

- 2632 • when a Device detects that communication with the Communications Hub is no longer
2633 available, it shall attempt to rejoin a HAN with the same Extended PAN ID as the
2634 Communications Hub it was previously communicating with. If that rejoin is
2635 unsuccessful, the Device shall attempt to establish a new connection with any

2636 Communications Hub which has the Permit Joining flag in the beacon set, using its
2637 hashed TC link key as the initial TC link key; and

- the Communications Hub shall reject any command to restore its CHF Device Log where the Extended PAN ID is not equal to the Communications Hub's HAN Interface's IEEE address. Where it accepts such a command, the Communications Hub shall set its Permit Joining bit for 600 seconds or until all restored Devices have joined. When a Device attempts to join, the Communications Hub shall treat the restored hashed TC link key for that Device as the initial TC link key, and the joining process shall continue accordingly.

The GSME shall set the *BillDeliveredTrailingDigit* attribute to the same value as *PriceTrailingDigit* in the *Price* cluster.

2647 In line with the SMETS requirement, the UnitOfMeasure parameter in the
2648 PublishTariffInformation command, sent to a GSME shall be 0x00 (kWh) as per the Message
2649 Templates for GCS01a and GCS01b, shall apply to the Block Threshold N parameter in the
2650 PublishBlockThresholds command in such Messages. Contrary to ZSE, the GSME shall
2651 undertake the necessary calculation when comparing these thresholds against the
2652 CurrentBlockPeriodConsumptionDelivered attribute (whose unit of measure in line with
2653 SMETS shall be m³).

2654 Contrary to ZSE, the GPF shall accept any valid UTCTime in the value of the
2655 Implementation Date/Time parameter in a Publish Change Of Tenancy command, in a
2656 Command complying with Use Case GCS09.

2657 10.5 Hand Held Terminal (HHT) interactions

2658 10.5.1 Introduction - informative

An HHT allows for delivery of Remote Party Messages to and from the SMHAN. This is as an alternative delivery route to the Communications Hub's WAN connection. It is intended for one-off configuration of Devices, for example at installation. Hence, there are time outs to ensure usage is limited in this way.

2663 This Section 10.5 specifies requirements related to:

- 2664 • how a connection is made between an HHT and a Communications Hub; and
2665 • how Remote Party Messages are then to be transferred to and from the HHT.

10.5.2 Establishing a connection between an HHT and a Communications Hub - informative

2668 The ZSE specification defines an *Inter-PAN Communications* mechanism. This mechanism
2669 is used to establish an initial secure link between the HHT and the Communications Hub,
2670 with the security being provided by ZSE's CBKE. Once this secure link is established:

- the HHT uses the link to send its Entity Identifier and *Install Code* to the Communications Hub;
 - the Communications Hub adds these details to the CHF's Device Log (so allowing the HHT to *join* the SMHAN); and
 - the HHT then *joins* the SMHAN and so can exchange Remote Party Messages within the Communications Hub, and the Communications Hub can relay them to / from the specified Device(s) on the HAN.

Both the *Inter-PAN Communications* and *joining* to the SMHAN use the CBKE mechanism that is defined in ZSE.

2680 *Inter-PAN Communications* shall only be available for 60 minutes from power on of the
2681 Communications Hub. So, if needed, *Inter-PAN Communications* can be enabled by power
2682 cycling the Communications Hub.

2683 The *Inter-PAN Communications* mechanism defined by ZSE requires the HHT to specify the
2684 Communications Hub that it wishes to link to. There may be multiple Communications Hubs
2685 available to the HHT to connect to via *Inter-PAN Communications*.

2686 There are a number of options to provide the HHT with information sufficient to identify
2687 uniquely the Communications Hub it is to link to, including:

- 2688 • the installer manually reading the GPF's Entity Identifier (which is the IEEE address of
2689 the Communications Hub's SMHAN radio) printed on the Hub, and confirming / selecting
2690 this on the HHT; or
- 2691 • the installer using a scanner on the HHT to read the GPF's Entity Identifier.

2692 Two illustrative connection scenarios are provided in the following two sections

2693 **10.5.2.1 Illustration 1: Installer manually chooses network - informative**

- 2694 42. the Communications Hub opens inter-PAN communication for 60 minutes after power on;
- 2695 43. the HHT is powered on;
- 2696 44. the HHT performs an active scan using the *Beacon Request* mechanism;
- 2697 45. the HHT displays the IEEE addresses returned in the *Beacons* from all neighbouring
2698 *PAN Coordinators*. Note that the GBCS requires the *Extended PAN ID* to be set to the
2699 Communications Hub's HAN Interface's IEEE address. This is the same as the GPF
2700 Entity Identifier, which is printed on the Communications Hub in line with CHTS;
- 2701 46. the installer (who knows the Consumer's Communications Hub's ZigBee IEEE address
2702 as the GPF Entity Identifier is printed on the Communications Hub) picks the desired
2703 IEEE address;
- 2704 47. the HHT initiates *Inter-PAN CBKE* with the Communications Hub;
- 2705 48. the Communications Hub responds to the *Inter-PAN CBKE*;
- 2706 49. if *Inter-PAN CBKE* completes successfully:
 - 2707 o The HHT sends its Install Code and Entity Identifier to the Communications Hub in a
2708 command secured using the shared symmetric key (the *APS link key*) produced by
2709 the *Inter-PAN CBKE* process; then
 - 2710 o The Communications Hub adds the HHT to the CHF Device Log; and then
 - 2711 o The HHT then *joins* to SMHAN.
- 2712 50. otherwise, no link is established.

2713 **10.5.2.2 Illustration 2: HHT uses barcode scan - informative**

- 2714 51. the Communications Hub opens inter-PAN communication for 60 minutes after power on;
- 2715 52. the HHT is powered on;
- 2716 53. the HHT optically scans the GPF Entity Identifier printed on the target Communications
2717 Hub;
- 2718 54. the HHT performs an active scan using the *Beacon Request* mechanism;
- 2719 55. when a Beacon returns an IEEE address equal to the scanned GPF Entity Identifier, the
2720 HHT initiates *Inter-PAN CBKE* with the Communications Hub so identified;
- 2721 56. the Communications Hub responds to the *Inter-PAN CBKE*;

- 2722 57. if *Inter-PAN CBKE* completes successfully:
- 2723 ○ the HHT sends its Install Code and Entity Identifier to the Communications Hub in a
2724 command secured using the shared symmetric key (the *APS link key*) produced by
2725 the *Inter-PAN CBKE* process; then
- 2726 ○ the Communications Hub adds the HHT to the CHF Device Log; and then
- 2727 ○ the HHT then *joins* to SMHAN.
- 2728 58. otherwise, no link is established.

10.5.3 WAN proxy operation

10.5.3.1 Introduction – informative

2731 The HHT has to be capable of holding Remote Party Messages, to which the appropriate
2732 Remote Party Message protection has already been applied, and has to be capable of
2733 exchanging such Messages.

2734 The Communications Hub must therefore be able to maintain two effective ‘WAN’ interfaces;
2735 the real one via the WAN network interface and a ‘logical WAN’ via the connection to the
2736 HHT.

10.5.3.2 WAN Responses

2738 The Communications Hub shall send any Responses and Alerts through both its WAN
2739 interface and the link to the HHT, if present. Whilst this may result in apparent unsolicited
2740 Responses at the Remote Party which have to be dealt with, it ensures the earliest possible
2741 reconciliation of Commands destined for Smart Metering Equipment.

10.5.3.3 HHT and CHF – Device Requirements

2743 As per Section 10.2.2, in all interactions between an HHT and a Communications Hub:

- 2744 ● the HHT shall support the *Tunneling Cluster* as a *Client*; and
- 2745 ● the Communications Hub shall support the *Tunneling Cluster* as a *Server*.

2746 The Communications Hub shall only allow *Inter-PAN Communications* for 60 minutes from
2747 any power on of the Communications Hub. For clarity, this is the period during which an
2748 HHT can establish a connection, not the period of use of any connection.

2749 At power on, a Communications Hub shall remove any Devices of type HHT (so a Device
2750 with device_type = 0x7E with its DLMS COSEM class_id 104 meaning) from the CHF
2751 Device Log.

2752 To exchange Remote Party Messages, the Communications Hub and HHT shall only use
2753 the *TransferData* command, and default responses to such commands.

2754 The Communications Hub shall always set *nwkExtendedPANId* to be the Entity Identifier of
2755 the GPF, which is always the Communications Hub’s IEEE address for its HAN Interface.

10.5.3.4 HHT and CHF – establishing communications

2757 Prior to being able to exchange Messages, the HHT and Communications Hub shall
2758 undertake the following steps:

- 2759 59. the HHT shall identify the Communications Hub and initiate the *CBKE* process using
2760 *Inter-PAN Communications*, as specified in the ZSE specification;
- 2761 60. the Communications Hub shall not respond to any such request if more than 60 minutes
2762 has elapsed since the Communications Hub’s most recent power on, or if there is a
2763 Device of type HHT already in the CHF’s Device Log. Otherwise, the Communications
2764 Hub shall respond to the *CBKE* request;

- 2765 61. if *CBKE* does not succeed, processing shall cease. Otherwise, processing shall
2766 continue from step 4;
- 2767 62. using the *APS link key* established through *CBKE* to secure the commands:
- 2768 h) the HHT shall send a *RequestTunnel* command to the Communications Hub, with
2769 contents as per Section 10.2.2;
- 2770 i) the Communications Hub shall send a *RequestTunnel/Response* command in
2771 response;
- 2772 j) if *TunnelStatus* in the response is not 0x00 ('success'), processing by the HHT shall
2773 cease. Otherwise the HHT shall send a *TransferData* command with the *TunnelID*
2774 parameter set to the *TunnelID* provided in the *RequestTunnel/Response* command
2775 and the *Data* parameter payload set to the concatenation:
2776 Entity Identifier of the HHT || 16 octet *Install Code* of the HHT
- 2777 k) on receipt of the *TransferData* command, the Communications Hub shall:
- 2778 ▪ add the HHT's Entity Identifier to the CHF Device log, recording the Device as
2779 being of type HHT, so with a device_type of 0x7E with its DLMS COSEM
2780 class_id 104 meaning;
- 2781 ▪ permit joining of the SMHAN for either (1) 240 seconds or (2) until the HHT has
2782 joined the SMHAN, whichever is the earlier; and
- 2783 ▪ start a timer. When that timer reaches 0xFFFF seconds, the CHF shall remove
2784 the HHT from its Device Log, remove the HHT from the SMHAN and close any
2785 open tunnels to the HHT.
- 2786 l) having added the HHT to its Device Log, the CHF shall send a *Default Response* to
2787 the HHT and close the tunnel to the HHT;
- 2788 m) on receipt of the *Default Response*, the HHT shall, if that *Default Response* contains
2789 a *Status Code* of 0x00 ('success'), attempt to *join* the SMHAN;
- 2790 n) if the *joining* is successful, the HHT shall send a *RequestTunnel* command to the
2791 CHF, with contents as per Section 10.2.2;
- 2792 o) the CHF shall process the *RequestTunnel* command and send a
2793 *RequestTunnel/Response* command in response;
- 2794 p) if *TunnelStatus* in the *RequestTunnel/Response* command is not 0x00 ('success'),
2795 processing by the HHT shall cease. Otherwise the HHT and CHF may now
2796 exchange Messages using the *TransferData* command.

2797 Note that steps 1 to 4.e) above use *Inter-PAN Communications*; the remaining steps use the
2798 standard ZigBee SMHAN communications.

2799 Once the HHT has *joined* the SMHAN, any Messages received by the CHF from the HHT in
2800 the *Data* parameter payload of a *TransferData* command, shall be forwarded to the relevant
2801 Device on the SMHAN as if they were received via the Communications Hub's WAN
2802 interface.

2803 Whilst the HHT is in the CHF's Device Log and *joined* to the SMHAN, any Responses
2804 received by the CHF from any SMHAN Device shall be provided to the HHT using the
2805 *TransferData* command. Such Responses shall also be sent over the Communications
2806 Hub's WAN interface, if available.

2807 Once the HHT usage on the SMHAN is complete, the HHT should send a *CloseTunnel*
2808 command to the Communications Hub. On receipt of such a *CloseTunnel* command from an

2809 HHT, the Communications Hub shall process that command as per the ZSE specification
2810 and shall:

- 2811 • remove the HHT from its Device Log; and
2812 • remove the HHT from the SMHAN.

11 Downloading firmware images to Devices

11.1 Introduction – informative

Compared to other Smart Metering messages, firmware images are large. Further, each image is likely to be applicable to a significant number of Devices. Thus, an end-to-end, unicast Message to each affected Device, with each message containing a copy of the image, is not efficient from a WAN perspective.

This leads to the approach for firmware update process being separated into two stages:

- distribution of the image to end Devices without any activation of that image; and
- a separate and subsequent ‘activation’ Command to each Device.

The Distribute Firmware Command is not a Critical Command (since it does not affect the operating firmware) and does not need to be unicast.

The Activate Firmware Command is a Critical Command and so must be unicast – as it must be digitally signed and be for one, and only one specified Device. Further, the Activate Command must apply to one, and only one, image and that image must have originated from the same party that signs the Activate Firmware Command (that is, the party responsible for that Device). To meet these requirements:

- the Activate Firmware Command is of type SME.C.C and so the Signature and MAC on the Command shall have been verified by the Device prior to the Hash validation (see next bullet); and
- a Device receiving an Activate Firmware Command shall calculate a Hash over the Manufacturer Image it holds and ensure the Hash so calculated matches that in the Activate Firmware Command, before the Device attempts to activate the corresponding Manufacturer Image.

The GBCS does not constrain the mechanisms used by Device manufacturers to ensure that only valid Manufacturer Images are activated on Devices manufactured by them. The GBCS does require that the manufacturer information related to a Manufacturer Image is made available, so that the Upgrade Image and the ZigBee Over-The-Air (OTA) Header can be provided when requesting distribution of an image.

In common with other Messages, the GBCS shall not constrain the mechanisms by which the firmware Messages are transported to the Communications Hub. The GBCS constrains HAN transport mechanisms to those provided by ZSE.

11.2 Common Requirements

11.2.1 Transport of firmware images

Italicised terms in this Section 11.2.1 shall have the meanings defined in ZigBee Document 09-5264-23.

For ESME and GSME firmware image distribution, the ZigBee Over-The-Air (OTA) mechanisms shall be used for transport of the image over the HAN. The ESME / GSME firmware image delivered to the Communications Hub shall comply with ZigBee OTA format requirements.

Communications Hub firmware images shall not be transported over the HAN and so ZigBee OTA structures shall not be required.

2855 Every Communications Hub shall be configured to act as the single OTA Server on its HAN.
2856 ESME and GSME shall be configured to act as an OTA Client. The ESME shall use the
2857 '*Image Notify*'¹⁹ *Command* sent by the OTA Server to inform it that a new firmware image is
2858 available. The GSME shall use the notification flags mechanism whereby a flag shall be set
2859 by the OTA Server to inform it that a new firmware image is available when requested.

2860 The Communications Hub shall:

- 2861 • as required by CHTS, have the capability to store one GSME OTA Upgrade Image and
2862 one ESME OTA Upgrade Image; and
- 2863 • overwrite an image with a subsequently delivered image for the same Device type
2864 unless:
 - 2865 ○ the subsequently delivered image has Force Replace = 0x00; and
 - 2866 ○ the Communications Hub has sent at least one *Image Block Response Command*
2867 relating to the already stored image but has not received a corresponding *Upgrade*
2868 *End Request Command*²⁰.

2869 In such circumstances the Communications Hub shall not overwrite the currently stored
2870 image.

2871 Whenever the Communications Hub's OTA Server issues an *Upgrade End Response*
2872 *Command* to a GSME or ESME pursuant to this GBCS, the *UpgradeTime* parameter shall
2873 have the value 0xFFFFFFFF²¹.

2874 The OTA Server shall not issue *Image Block Response Commands* with WAIT_FOR_DATA
2875 status.

2876 Contrary to Section 6.13 of ZigBee Document 09-5264-23, the OTA Client shall not activate
2877 the OTA Image except as specified in Use Case CS06.

2878 **11.2.2 Construction of Upgrade Image**

2879 For an ESME or GSME firmware image, the Authorising Remote Party shall be the Supplier
2880 for the target Device.

2881 For a Communications Hub firmware image, the Authorising Remote Party shall be the WAN
2882 Provider for the target Device.

2883 Upgrade Image shall be the concatenation:

2884 Manufacturer Image || Force Replace || 0x40 || Authorising Remote Party Signature

2885 where:

- 2886 • Manufacturer Image shall contain the firmware image the Device is to apply and any
2887 manufacturer specific data needed. For clarity, the GBCS shall not constrain the
2888 structure or contents of Manufacturer Image;
- 2889 • Force Replace shall be a single octet where Force Replace = 0x00 shall mean do not
2890 force the replacement of the currently stored image; and
- 2891 • Authorising Remote Party Signature shall be calculated across the Manufacturer Image
2892 using the Authorising Remote Party's Private Digital Signing Key.

2893 **11.2.3 Construction of OTA Upgrade Image**

2894 OTA Upgrade Image shall be the concatenation:

¹⁹ See Section 6.10.3 of ZigBee Document 09-5264-23

²⁰ As defined in Section 6.10 of ZigBee Document 09-5264-23

²¹ As defined in Sections 6.10.10 and 6.8.4 of ZigBee Document 09-5264-23

2895 OTA Header || Upgrade Image

2896 where OTA Header shall be populated according to Table 11.2.3. For clarity, there shall be
2897 no other sub-elements present.

OTA Header			
ZigBee OTA Message Element	Contents	Length (octets)	Note
OTA upgrade file identifier	0x0BEEF11E	4	Fixed by ZigBee OTA specification
OTA Header version	0x0100	2	Specified by current version of ZigBee OTA specification
OTA Header length	0x003C	2	The length of ZigBee OTA Header which is decimal 60
OTA Header Field control	0x0004	2	Detailed what is / is not present in ZigBee OTA Header
Manufacturer code	ZSE assigned identifier for the Manufacturer of the target Device	2	So this identifies the manufacturer producing the Manufacturer Image
Image type	Manufacturer specific	2	As per the ZigBee OTA specification, this is to differentiate products from the same manufacturer
File version	Manufacturer specific	4	As per the ZigBee OTA specification, this is to differentiate release and build numbers for the product in question
ZigBee Stack version	0x0002	2	ZigBee PRO
OTA Header string	Manufacturer specific	32	May be blank but is not required to be used in Device processing of the firmware image
Total Image size (including header)	The length in octets of OTA Upgrade Image	4	Contents to be interpreted as an unsigned integer
Minimum hardware version	Manufacturer specific	2	
Maximum hardware version	Manufacturer specific	2	

2898 Table 11.2.3: Population of the OTA Header The OTA Header shall uniquely identify a
2899 firmware image.

2900 **11.2.4 Construction of Manufacturer Image Hash**

2901 Manufacturer Image Hash shall be a Hash calculated across the whole Manufacturer Image
2902 file that is provided to the Authorising Remote Party.

2903 **11.2.5 Verification of the authenticity of the Upgrade Image**

2904 The Device shall verify Upgrade Image by verifying the Authorising Remote Party Signature
2905 using Manufacturer Image and the Authorising Remote Party's Public Key. For clarity, this
2906 shall be the only ECDSA verification required by the GBCS and this is not the ZSE ECDSA
2907 Signature sub-element.

2908 For an ESME or GSME receiving an Upgrade Image, the Authorising Remote Party's Public
2909 Key shall be held by the Device in the {supplier, digitalSignature,
2910 management} Trust Anchor Cell.

2911 For a Communications Hub receiving an Upgrade Image, the Authorising Remote Party's
 2912 Public Key shall be that held by the Device in the {wanProvider, digitalSignature,
 2913 management} Trust Anchor Cell.

2914 **11.2.6 Construction of Firmware Distribution Receipt Alert**

2915 If the Device is an ESME or a Communications Hub, the 'Alert Payload' fields shall be
 2916 populated according to Section 7.2.9.

2917 If the Device is a GSME, the 'Alert Payload' fields shall be populated according to Section
 2918 7.2.10.

2919 In all cases, the Device shall:

- 2920 • populate the Use Case Specific Additional Content with the concatenation
 2921 0x0940 || the calculated Manufacturer Image Hash
- 2922 • populate the Alert Code field with 0x801C (failure), or 0x8072 (success).

2923 **11.2.7 Activation of firmware images**

2924 The Activate Firmware Command shall be of type SME.C.C.

2925 A Device receiving such a Command shall undertake the verifications required of a SME.C.C
 2926 Command.

2927 If all such SME.C.C verifications succeed, the Device shall then calculate Manufacturer
 2928 Image Hash over the Manufacturer Image it holds and compare that with the Manufacturer
 2929 Image Hash specified in the Activate Firmware Image Command (see Use Case CS06 in
 2930 Section 11.5 for details of the Activate Firmware Command Payload construction).

2931 If the two Hashes match, the Device shall attempt to activate the firmware image.

2932 If the two Hashes do not match, the Device shall not attempt to activate the firmware image.

2933 The Device shall issue a relevant Activate Firmware Image Response detailing the success
 2934 or failure (see Use Case CS06 in Section 11.5 for details of the Activate Firmware Command
 2935 Payload construction).

2936 **11.3 CS05a Distribute Firmware to Communications Hub**

2937 This Use Case covers the distribution of an Upgrade Image that is intended for a
 2938 Communications Hub to that Communications Hub.

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Command and Response
Message Type Category	None – this is a Variant Message
Capable of future dated invocation?	No
Protection Against Replay Required?	No
SEC User Gateway Services Schedule (Service Request) Reference	N/A (this Command is not available as part of the User Gateway Services Schedule)
Valid Target Device(s)	Communications Hub
Valid Business Originator role(s) for Command invocation (and so, for DLMS COSEM Commands, which Application Association	WAN Provider

is to be used for delivery of the Command to the Device) [Remote Party Messages Only]	
Valid Response Recipient role(s) (only for Messages authorised by the Access Control Broker on behalf of parties not known to the Device) [Remote Party Messages Only]	N/A
Valid initiating Device type(s) [HAN Only Messages]	N/A
Protocol	CSP Specific

2939 Table 11.3: Use Case Cross References for CS05a Distribute Firmware to Communications Hub

11.3.1 Pre-conditions

2941 None.

11.3.2 Detailed Steps

2943 The Upgrade Image shall be constructed according to Section 11.2.2.

2944 The Upgrade Image shall be transported to the Communications Hub.

2945 The Communications Hub shall verify the Upgrade Image according to Section 11.2.5, verify
2946 the Upgrade Image is suitable for this Communications Hub, and update its Event Log with
2947 the outcome of that verification.

2948 If the verification is successful, the Communications Hub shall construct and send a
2949 Firmware Distribution Receipt Response, according to Section 11.2.6 and shall store the
2950 Manufacturer Image contained within the Upgrade Image.

2951 If the verification is not successful, the Communications Hub shall discard the upgrade
2952 image and construct and send a Firmware Distribution Receipt Alert, according to Section
2953 11.2.6.

2954 On receipt of a Firmware Distribution Receipt Response, the WAN Provider may verify the
2955 cryptographic protection as specified in Section 6.8.3.

2956 Additionally, the WAN Provider may verify that the Manufacturer Image received is that
2957 intended by comparing the Manufacturer Image Hash in the Firmware Distribution Receipt
2958 Response, with the Hash which it calculates over the Manufacturer Image provided.

11.4 CS05b Distribute Firmware to GSME / ESME

2960 This Use Case covers the distribution of an OTA Upgrade Image that is intended for a
2961 GSME or ESME to that GSME or ESME.

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Command and Response
Message Type Category	None – this is a Variant Message
Capable of future dated invocation?	No
Protection Against Replay Required?	No
SEC User Gateway Services Schedule (Service Request) Reference	11.1
Valid Target Device(s)	ESME / GSME -
Valid Business Originator role(s) for Command invocation (and so, for DLMS COSEM Commands, which Application Association	Supplier

is to be used for delivery of the Command to the Device) [Remote Party Messages Only]	
Valid Response Recipient role(s) (only for Messages authorised by the Access Control Broker on behalf of parties not known to the Device) [Remote Party Messages Only]	N/A
Valid initiating Device type(s) [HAN Only Messages]	N/A
Protocol	CSP Specific to Communications Hub; ZigBee OTA from Communications Hub to ESME / GSME

Table 11.4: Use Case Cross References for CS05b Distribute Firmware to ESME / GSME

11.4.1 Pre-conditions

None.

11.4.2 Detailed Steps

Italicised terms in this Section 11.4.2 shall have the meaning specified in ZSE.

ESME and GSME shall use the ZCL *Image Block Request* and *Image Block Response* commands to retrieve available OTA images.

ESME and GSME shall not use the ZCL *Query Specific File Request* and *Query Specific File Response* commands.

The OTA Upgrade Image shall be populated according to Section 11.2.3.

The OTA Upgrade Image shall be transported to the Communications Hub through which the Device communicates.

The Communications Hub shall update its OTA Cluster to reflect availability of the OTA Upgrade Image, once the image is received by the Communications Hub.

The Communications Hub, as OTA Server, shall indicate availability of an OTA Upgrade Image differently for ESME and GSME:

- for ESME, the Communications Hub shall send a ZSE *Image Notify* command; and
- for GSME, the Communications Hub shall set a *the New OTA Firmware flag* (Bit Number 0) in *FunctionalNotificationFlags*.

The ESME / GSME shall download an OTA Upgrade Image when it is aware of the availability of a suitable OTA Upgrade Image using the *QueryNextImage* and *Image Block/Page* commands specified in the OTA Cluster specification²².

The ESME / GSME shall verify the Upgrade Image contained within the OTA Upgrade Image according to Section 11.2.5, and update its Event Log with the outcome of that verification.

If the verification is successful, the ESME / GSME shall construct and send a Firmware Distribution Receipt Alert, according to Section 11.2.6, and shall store the Manufacturer Image contained within the OTA Upgrade Image.

If the verification is not successful, the Device shall discard the OTA Upgrade Image, and send a Firmware Verification Failed Alert, as detailed in Section 11.3.2.

On receipt of a Firmware Distribution Receipt Alert, the Supplier may verify the cryptographic protection as specified in Section 6.8.3.

²² ZigBee Document 095264

2993 Additionally, the Supplier may verify that the Manufacturer Image received by the Device is
 2994 that intended by comparing the Manufacturer Image Hash in the Firmware Distribution
 2995 Receipt Response, with the Hash which it calculates over the Manufacturer Image provided.

2996 11.5 CS06 Activate Firmware

2997 This Use Case covers the activation of a Firmware Image.

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Command, Response and Alert (if future dated)
Message Type Category	SME.C.C
Capable of future dated invocation?	Yes
Protection Against Replay Required?	No
SEC User Gateway Services Schedule (Service Request) Reference	N/A for Communications Hub (this Command is not available as part of the User Gateway Services) 11.3 for ESME and GSME
Valid Target Device(s)	ESME / GSME / CH
Valid Business Originator role(s) for Command invocation (and so, for DLMS COSEM Commands, which Application Association is to be used for delivery of the Command to the Device) [Remote Party Messages Only]	Supplier for ESME / GSME WAN Provider for CH
Valid Response Recipient role(s) (only for Messages authorised by the Access Control Broker on behalf of parties not known to the Device) [Remote Party Messages Only]	N/A
Valid initiating Device type(s) [HAN Only Messages]	N/A
Protocol	ASN.1 -

2998 Table 11.5: Use Case Cross References for CS06 Activate Firmware

2999 11.5.1 Pre-conditions

3000 None.

3001 11.5.2 Detailed Steps

3002 11.5.2.1 Construction of Command

3003 Activate Firmware Command Payloads shall be constructed according to the requirements
 3004 of Section 11.5.2.3 and populated as specified in Table 11.5.2.1.

3005 MAC Header, Grouping Header, KRP Signature and ACB-SMD MAC shall be populated as
 3006 required for a Command of the SME.C.C Message Category.

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
@ActivateFirmware.CommandPayload	SEQUENCE			

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
manufacturerImageHash	OCTET STRING	The Manufacturer Image Hash of the image to be activated.	Mandatory	An octet-string of length 32 interpreted as the Manufacturer Image Hash of the Manufacturer Image that is to be activated
executionDateTime	Generalized Time	The date-time at which the Command is to be executed, if future dated	OPTIONAL	

3007 Table 11.5.2.1: @ActivateFirmware.CommandPayload population

3008 **11.5.2.2 Device processing of Command and Response handling**

3009 The Device receiving an Activate Firmware Command shall undertake processing steps in
3010 the sequence defined in this Section 11.5.2.2.

3011 The Device shall:

3012 63. undertake Command Authenticity and Integrity Verification as required for a Command of
3013 the SME.C.C Message Category;

3014 64. if executionDateTime is present then the Device shall:

- 3015 o record manufacturerImageHash, originatorCounter and
3016 executionDateTime;

- 3017 o construct and send a Response where executionOutcome is not present.
3018 Grouping Header is constructed and Response Cryptographic Protection is applied
3019 as required for a Response of the SME.C.C Message Categories; and

- 3020 o at the date-time specified in executionDateTime, undertake the processing from
3021 step 3.

3022 If executionDateTime is not present then the Device shall continue processing from
3023 step 3 immediately;

3024 65. if the Device does not have a stored Manufacturer Image then set
3025 activateImageResponseCode to noImageHeld and process from step 7;

3026 66. calculate Manufacturer Image Hash. If the calculated value does not equal
3027 manufacturerImageHash then the Device shall set activateImageResponseCode
3028 to hashMismatch and process from step 7;

3029 67. attempt to activate Manufacturer Image. If the activate fails then the Device shall set
3030 activateImageResponseCode to activationFailure and process from step 7;

3031 68. set activateImageResponseCode to success ;

3032 69. populate the executionOutcome according to the requirements of Section 11.5.2.3
3033 using the activateImageResponseCode value produced by the processing in this
3034 Section 11.5.2.2, the value of originatorCounter from the Command and the
3035 version of firmware now in operation to populate firmwareVersion;

3036 70. construct Grouping Header and apply the Response Cryptographic Protection required
3037 for a Response / Alert of the SME.C.C / SME.A.C Message Categories respectively. In
3038 such an Alert, the Message Code shall be 0x00CA. The Response / Alert shall be
3039 addressed to the Business Originator of the Corresponding Command. If

3040 activateImageResponseCode **is** success **then** alertCode **shall be** 0x0066 **else**
 3041 alertCode **shall be** 0x0067; **and**

3042 71. **send the Response if** executionDateTime **was not present in the Command or send**
 3043 **the Alert if** executionDateTime **was present in the Command.**

3044 On receipt of the Response, the recipient may undertake the 'Response Recipient
 3045 Verification' for Responses of type SME.C.C. or for Alerts of type SME.A.C, dependent upon
 3046 the Message received.

3047 **11.5.2.3 Activate Firmware Command, Response and Alert Payloads - structure 3048 definition**

3049 **Each instance of** @ActivateFirmware.CommandPayload **and of**
 3050 @ActivateFirmware.ResponsePayload **and of**
 3051 @ActivateFirmware.AlertPayload **shall be an octet string containing the DER**
 3052 **encoding of the populated structure defined in this Section 11.5.2.3 which specifies the**
 3053 **structure in ASN.1 notation.**

```
3054     ActivateFirmware DEFINITIONS ::= BEGIN
3055
3056     CommandPayload ::= SEQUENCE
3057     {
3058        -- specify the hash of the Manufacturer Image to be activated
3059        manufacturerImageHash OCTET STRING,
3060
3061        -- the Originator Counter as in the Grouping Header of the Command
3062        originatorCounter INTEGER (0..9223372036854775807),
3063
3064        -- the date-time at which the Command is to execute, if future dated
3065        executionDateTime GeneralizedTime OPTIONAL
3066     }
3067
3068     ResponsePayload ::= CHOICE
3069     {
3070        -- if the Command is future dated, the Response will not have any details of
3071        -- execution (those will be in the subsequent alert)
3072        commandAccepted NULL,
3073
3074        -- if the Command is for immediate execution, the Response will detail the
3075        -- outcomes
3076        executionOutcome ExecutionOutcome
3077     }
3078
3079     AlertPayload ::= SEQUENCE
3080     {
3081        -- specify the Alert Code
3082        alertCode INTEGER(0..4294967295),
3083
3084        -- specify the date-time of execution
3085        executionDateTime GeneralizedTime,
3086
3087        -- the Originator Counter as in the Grouping Header of the corresponding Command
3088        originatorCounter INTEGER (0..9223372036854775807),
3089
3090        -- detail what happened when the future dated command was executed
3091        executionOutcome ExecutionOutcome
3092     }
3093
3094     ExecutionOutcome ::= SEQUENCE
3095     {
3096        -- Specify whether the activation was successful or not
3097        activateImageResponseCode ActivateImageResponseCode,
3098 }
```

```
3099      -- Specify the Device's now current firmware version
3100      firmwareVersion                         OCTET STRING
3101  }
3102
3103  ActivateImageResponseCode ::= INTEGER
3104  {
3105      success                               (0),
3106      noImageHeld                           (1),
3107      hashMismatch                          (2),
3108      activationFailure                    (3)
3109  }
3110
3111 END
```

12 Requirements for Certificates

This Section 12 lays out requirements as to structure and content to which all valid authorised Certificates shall comply, in so far as those requirements affect the processing carried out by Devices. All terms in this section shall, where not defined in the GBCS, have the meanings in IETF RFC 5759²³ and IETF RFC 5280.

12.1 Requirements applicable to all Certificates

All Security Credential Documents that are successfully authorised within the APKI for use by Devices within the scope of this GBCS shall:

- be compliant with IETF RFC 5759 and so with IETF RFC 5280. In adherence with the requirements of IETF RFC5759, all Security Credential Documents shall:
 - contain the authorityKeyIdentifier extension, except where the Security Credential Document is self-signed;
 - contain the keyUsage extension which shall be marked as critical;
- be X.509 v3 certificates as defined in IETF RFC 5280, encoded using the ASN.1 Distinguished Encoding Rules;
- only contain public keys of types that are explicitly allowed within the GBCS. This means all public keys shall be elliptic curve public keys on the NIST P-256 curve;
- only contain public keys in uncompressed form which shall be elliptic curve points in uncompressed form as detailed in Section 2.2 of IETF RFC 5480²⁴;
- only provide for signature methods that are explicitly allowed within the GBCS. This means using P-256 Private Keys with SHA 256 and ECDSA;
- contain a serialNumber of no more than 8 octets in length;
- contain a subjectKeyIdentifier which shall be marked as non-critical;
- contain a certificatePolicies extension containing at least one PolicyIdentifier which shall be marked as critical. For clarity and in adherence with IETF RFC 5280, Certification Path Validation undertaken by Devices shall interpret this extension;
- contain an authorityKeyIdentifier in the form [0] KeyIdentifier which shall be marked as non-critical, except where the Security Credential Document is self-signed. Note this exception only applies where RemotePartyRole as specified in the X520OrganizationalUnitName field = root;
- only contain KeyIdentifiers generated as per method (2) of Section 4.2.1.2 of IETF RFC 5280. Thus KeyIdentifiers shall always be 8 octets in length;
- contain an IssuerName which is identical to the Security Credential Document's signer's SubjectName; and
- have a valid notBefore field consisting of the time of issue encoded and a valid notAfter as per IETF RFC 5280 Section 4.1.2.5.

²³ <http://tools.ietf.org/html/rfc5759>

²⁴ <http://tools.ietf.org/html/rfc5480>

3149 12.2 Requirements applicable to Organisations' 3150 Certificates only

3151 All Organisations' Certificates that are Authorised for use by Devices within the scope of this
3152 GBCS shall:

- 3153 • have a fixed expiration date in the `notAfter` field which shall not be
3154 GeneralizedTime value of 99991231235959Z;
- 3155 • contain a non-empty subject field which shall contain a unique X.500 Distinguished
3156 Name (DN), which shall be the unique trading name of the Organisation, and an
3157 `X520OrganizationalUnitName` whose value shall be set to the `RemotePartyRole`
3158 that this Certificate allows the subject of the Certificate to perform; and
- 3159 • contain a single Public Key except where the `RemotePartyRole` = `root`. Where the
3160 `RemotePartyRole` = `root`, the Certificate shall contain two public keys. The second
3161 public key shall be referred to as the Contingency Key²⁵ and shall be present in the
3162 `WrappedApexContingencyKey` extension with the meaning of IETF RFC 5934²⁶. The
3163 Contingency Key shall be Encrypted as per the requirements of Section 13.3.5.8.1.

3164 12.3 Requirements applicable to Certificates where 3165 `RemotePartyRole` = `root` or `issuingAuthority`

3166 All Remote Parties' Certificates that:

- 3167 • are Authorised within the APKI for use by Devices within the scope of this GBCS; and
- 3168 • have a `X520OrganizationalUnitName` whose value is either `root` or
3169 `issuingAuthority`

3170 shall:

- 3171 • have a `keyUsage` with a value of `keyCertSign` and `cRLSign`. For clarity, Devices
3172 are not required to use the associated Public Keys in relation to the `cRLSign`
3173 `keyUsage`;
- 3174 • where `X520OrganizationalUnitName` = `issuingAuthority`:
 - 3175 ○ contain at least one `policyIdentifier` in the `certificatePolicies` extension
3176 that refers to the OID(s) valid for usage in GB Smart Metering;
 - 3177 ○ contain the `basicConstraints` extension, with values `cA=True`, and `pathLen=1`.
3178 This extension shall be marked as critical;
- 3179 • where `X520OrganizationalUnitName` = `root`:
 - 3180 ○ contain a single `policyIdentifier` in the `certificatePolicies` extension that
3181 refers to the OID for `anyPolicy`; and
 - 3182 ○ contain the `basicConstraints` extension, with the value `cA=True` and `pathLen`
3183 absent (unlimited). This extension shall be marked as critical.

²⁵ The Contingency Key is a second public key held in the Root Certificate (and protected with an encryption key). Its sole purpose is to allow the validation of a specific command that allows direct replacement of the Root Trust Anchor. The command (an Apex Trust Anchor Update message) is signed with a private key (used once only, and only to sign this message) that only the second public key (known as the Contingency Key) can verify and therefore authorise action of.

²⁶ Housley, R., Ashmore, S., and C. Wallace, "Trust Anchor Management Protocol (TAMP)", RFC 5934, August 2010.
<https://tools.ietf.org/html/rfc5934>

3184 12.4 Requirements applicable to Certificates where 3185 RemotePartyRole is neither root nor 3186 issuingAuthority

3187 All Remote Parties' Certificates that:

- 3188 • are Authorised within the APKI for use by Devices within the scope of this GBCS; and
- 3189 • have a X520OrganizationalUnitName whose value is not root and is not
3190 issuingAuthority

3191 shall:

- 3192 • contain a subjectUniqueID whose value shall be the 8 octet Entity Identifier of the
3193 subject of the Certificate;
- 3194 • have a keyUsage with a value of only one of digitalSignature or keyAgreement;
3195 and
- 3196 • contain a single policyIdentifier in the certificatePolicies extension that
3197 refers to the OID applicable to the environment the Certificate has been issued in.

3198 12.5 Requirements applicable to Device Certificates

3199 All Device Certificates that are Authorised within the APKI for use by Devices within the
3200 scope of this GBCS shall:

- 3201 • not have a well-defined expiration date and so the notAfter field shall be assigned the
3202 GeneralizedTime value of 99991231235959Z;
- 3203 • have an empty SubjectName;
- 3204 • have a keyUsage with a value of only one of digitalSignature or keyAgreement;
- 3205 • contain a single policyIdentifier in the certificatePolicies extension that
3206 refers to the OID applicable to the environment the Device Certificate has been issued
3207 in;
- 3208 • contain a SubjectAlternativeName extension which shall contain a single
3209 GeneralName of type OtherName that is further sub-typed as a
3210 HardwareModuleName (id-on-HardwareModuleName) as defined in IETF RFC
3211 4108²⁷. The hwSerialNum field shall be set to the Device's Entity Identifier. In
3212 adherence to IETF RFC 5280, SubjectAlternativeName shall be marked as critical;
3213 and
- 3214 • contain a single Public Key.

3215 12.6 Device processing of Certificates

3216 In relation to Certificates, Devices shall:

- 3217 • accept unexpected (not required by the GBCS) certificate extensions and shall ignore
3218 silently non-critical unrecognized certificate extensions;
- 3219 • in adherence with the requirements of IETF RFC 5280, reject any certificate containing
3220 unrecognized critical certificate extensions; and
- 3221 • reject any certificate containing either policy mappings or name constraints.

²⁷ <http://tools.ietf.org/html/rfc4108>

13 Managing Security Credentials on Devices

13.1 Introduction - informative

This Section 13 includes the Use Cases related to the management of Security Credentials on Devices in terms of the relevant Commands, Responses and Alerts:

- 13.2 - CS02a Provide Security Credential Details Command and Response;
- 13.3 - CS02b Update Security Credentials Command, Response and Alert;
- 13.4 - CS02c Issue Security Credentials;
- 13.5 - CS02d Update Device Certificates on Device;
- 13.6 - CS02e Provide Device Certificates from Device;
- 13.7 - Pair-wise Authorisation of Devices (covered by various Join / Unjoin Use Cases); and
- 13.8 - GPF Device Log Backup and Restore (GCS59 and GCS62).

13.1.1 Device Security Credentials - informative

In terms of processing relating to a Device's own Security Credentials:

- the Command to Devices for issuing Device Certificate Signing Requests (and therefore generate new Public-Private Key Pairs) is covered in Section 13.4;
- the Command to Devices for the Device to replace its current Device Certificate with a new Device Certificate resulting from a Device Certificate Signing Request is covered in Section 13.5, as are the related requirements for the capability to store such Documents; and
- the Command to a Device to provide a copy of its currently held Device Certificates is covered Section 13.6.

13.1.2 Remote Party Security Credentials - informative

This Section 13.1.2 summarises the GBCS requirements in relation to storing, replacing and providing details of Remote Party Security Credentials. The use of such credentials to control access to Device functions is detailed in other sections of the GBCS and in relevant Use Cases.

A Remote Party Security Credential is a Public Key Certificate which securely binds together the Remote Party's identity with a Public Key along with related information, including what that Public Key can be used for and over what time period it is valid. The corresponding Private Key should be securely controlled solely by the Remote Party and known only to that Remote Party.

The purpose of storing each Remote Party Public Key (and related details) on a Device is so that each Public Key can act as a 'Trust Anchor' for the Device. The Device uses these Trust Anchors to check cryptographically whether Remote Party Messages can be trusted or not (and so whether it should act on them or not). Thus, all of a Device's Trust Anchors must be populated.

Trust Anchors need to be capable of being replaced during a Device's operational life for a number of reasons including:

- the Certificate's expiry (Organisation Certificates will only be valid for a fixed period of time);
- the Known Party transferring control to a different organisation (for example on Change of Supplier);
- the cryptographic algorithms, or parameters such as key length, needing to be changed;
- the Known Party having lost the use of the corresponding Private Key; or
- there being concerns that someone other than the Known Party has use of, or may have use of, the corresponding Private Key.

Thus, an 'Update Security Credentials Command' must be supported by all Devices that rely on Remote Party Security Credentials to act as Trust Anchors. Related, all such Devices need to support a 'Provide Security Credential Details' Command, so that Remote Parties can be sure which Devices need to have credentials replaced.

However, if these Trust Anchors could be replaced without proper protections, attackers could take over control of Devices or the Devices could be rendered inoperable. Thus, a Device needs to do thorough checks before applying an Update Security Credentials Command. The checks that the Device can and must do vary dependent on the reasons for the change. Thus, Section 13.2.1 lays out a number of different checks and the circumstances in which corresponding Commands may be issued. Broadly the following checks are carried out by the Device:

- is the Command properly formed?
- is the Command for the Device that it has been delivered to, and is the Command one that it has not processed previously?
- are the Remote Parties apparently authorising the Command allowed to authorise it?
- was the Command Authorised by the Remote Parties that it appears to be Authorised by?
- were the Certificates in the payload of the Command issued by properly Authorised parties, specifically by Certification Authorities Authorised (by 'root' under the APKI) to issue GB Smart Metering Certificates?

Only when a Device has successfully undertaken all five sets of checks should it action the Update Security Credentials Command.

Other Critical Commands only have to complete the first four categories of check.

13.1.3 Trust Anchor Management (TAMP) - informative

The GBCS does not specify a fully compliant TAMP solution due to the limited processing and networking capability of Devices. However, it does incorporate checks that are functionally derived from relevant checks in IETF RFC 5934.

The GBCS only permits a restricted subset of 'IETF RFC 5934 like' functionality:

- replacement of Trust Anchors is required (and specified in this Use Case) but their addition, change or removal is not allowed;
- status queries are supported (and are specified in this Section 13.1.3); and
- community related functions are not supported.

3301 **13.2 CS02a Provide Security Credential Details Command**
 3302 **and Response**

3303 **13.2.1 Description**

3304 This section covers the creation, validation and processing of (i) **Provide Security Credential**
 3305 **Details Commands** and (ii) **Responses** to such **Commands**.

3306 **13.2.2 Use Case Cross References**

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Command and Response
Message Type Category	Variant Message and is not a Critical Command
Capable of future dated invocation?	No
Protection Against Replay Required?	No
SEC User Gateway Services Schedule (Service Request) Reference	6.24
Valid Target Device(s)	ESME / GSME / GPF / CHF / HCALCS / PPMID -
Valid Business Originator role(s) for Command invocation (and so, for DLMS COSEM Commands, which Application Association is to be used for delivery of the Command to the Device) [Remote Party Messages Only]	Supplier Network Operator Access Control Broker Transitional Change of Supplier WAN Provider Recovery
Valid Response Recipient role(s) (only for Messages Authorised by the Access Control Broker on behalf of parties not known to the Device) [Remote Party Messages Only]	N/A
Valid initiating Device type(s) [HAN Only Messages]	N/A
Protocol	ASN.1

3307 Table 13.2.2: Use Case Cross References for Provide Security Credential Details Command and
 3308 Response

3309 **13.2.3 Common Requirements**

3310 **13.2.3.1 Summary - informative**

3311 Remote Party Security Credentials are provided to Devices as Certificates which are X.509
 3312 based, DER encoded ASN.1 structures. Hence, the Command's structure is specified using
 3313 ASN.1 with DER encoding to be applied to Command instances. Note that the details
 3314 provided in the Response include the related Protection Against Replay counter details held
 3315 on the Device.

3316 13.2.3.2 The ‘Provide Security Credential Details’ Command and Response

3317 This Section 13.2.3.2 summarises the structure of the Provide Security Credential Details Command.

3318 If protected by an Access Control Broker MAC as per Section 13.2.4.2, a Provide Security Credential Details Command shall be the
3319 concatenation:

3320 MAC Header || Grouping Header || @ProvideSecurityCredentialDetails.Command || 0x00 || ACB-SMD MAC

3321 If protected by a KRP Signature as per Section 13.2.4.2, a Provide Security Credential Details Command shall be the concatenation:

3322 Grouping Header || @ProvideSecurityCredentialDetails.Command || 0x40 || KRP Signature

3323 If an SMD Signature is required as per Section 13.2.4.5, a Provide Security Credential Details Response shall be the concatenation:

3324 Grouping Header || @ProvideSecurityCredentialDetails.Response || 0x40 || SMD Signature

3325 If an SMD Signature is not required as per Section 13.2.4.5, a Provide Security Credential Details Response shall be the concatenation:

3326 MAC Header || Grouping Header || @ProvideSecurityCredentialDetails.Response || 0x00 || SMD-KRP MAC

3327 Where:

3328 • @ProvideSecurityCredentialDetails.Command and Response shall each be an octet string containing the DER encoding of the
3329 populated ASN.1 structure (as laid out in Section 13.2.3.3);

3330 • 0x40 is the length in octets of Signature when a SMD or KRP Signature is present, and 0x00 is the length in octets of Signature when a
3331 SMD or KRP Signature is not present;

3332 • KRP Signature and ACB-SMD MAC are as defined in Section 13.2.4.2;

3333 • SMD Signature and SMD-KRP MAC are as defined in Section 13.2.4.5; and

3334 • MAC Header and Grouping Header are as defined in Section 7.2.

3335 13.2.3.3 The @ProvideSecurityCredentialDetails.Command and @ProvideSecurityCredentialDetails.Response structure definition

3336 Each instance of @ProvideSecurityCredentialDetails.Command and of @ProvideSecurityCredentialDetails.Response shall
3337 be an octet string containing the DER²⁸ encoding of the populated structure defined in this Section 13.2.3.3 which specifies the structure in
3338 ASN.1 notation²⁹.

²⁸ <https://www.itu.int/rec/T-REC-X.690/en>

²⁹ <https://www.itu.int/rec/T-REC-X.680/en>

```
3339 ProvideSecurityCredentialDetails DEFINITIONS ::= BEGIN
3340
3341 Command ::=                               SEQUENCE
3342 {
3343   -- Identify which of the Public Keys on the Device is to be used in verifying the Signature or MAC
3344   -- (so defining the nature of the verification by way of the KeyUsage parameter held on the
3345   -- Device for the Public Key so identified).
3346
3347 authorisingRemotePartyTACellIdentifier      TrustAnchorCellIdentifier,
3348
3349   -- List the Remote Party Role(s) for which credential details are required
3350
3351 remotePartyRolesCredentialsRequired        SEQUENCE OF RemotePartyRole
3352 }
3353
3354 Response ::=                           SEQUENCE OF RemotePartyDetails
3355
3356 RemotePartyDetails ::=                  SEQUENCE
3357 {
3358
3359   -- Which Remote Party do these details relate to?
3360   remotePartyRole                      RemotePartyRole,
3361
3362   -- statusCode shall be success unless the role is not valid on this type of Device or there is a processing failure
3363   statusCode                           StatusCode,
3364
3365
3366   -- What is the current Update Security Credentials Protection Against Replay number on the Device for this role, where there is
3367   -- such a number for this role?
3368
3369   currentSeqNumber                     SeqNumber OPTIONAL,
3370
3371   -- What are the details held on the Device for each of the Cells related to this role? The list shall have between one and
3372   -- three entries (e.g. there will be one if role is transitional change of supplier; there may be three if role is supplier)
3373
3374   trustAnchorCellsDetails             SEQUENCE OF TrustAnchorCellContents OPTIONAL
3375 }
3376
3377 SeqNumber ::=                         INTEGER (0..9223372036854775807)
3378
3379 TrustAnchorCellContents ::=          SEQUENCE
3380 {
3381   -- To what cryptographic use can the Public Key in this Cell be put? Some Remote Party Roles
```

```
3382 -- (e.g. supplier) can have more than one Public Key on a Device and each one would only have
3383 -- a single cryptographic use.
3384
3385 trustAnchorCellKeyUsage           KeyUsage,
3386
3387 -- trustAnchorCellUsage is to allow for multiple Public Keys of the same keyUsage for the same Remote
3388 -- Party Role. This will be absent except where used to refer to the Supplier Key Agreement Key.
3389 -- This Key is used solely in relation to validating Supplier generated MACs on Prepayment Top Up transactions.
3390
3391 trustAnchorCellUsage           CellUsage DEFAULT management,
3392
3393 -- The subjectUniqueID which shall be the 64 bit Entity Identifier of the Security Credentials in this Trust Anchor Cell.
3394
3395 existingSubjectUniqueID          OCTET STRING,
3396
3397 -- The APKI requirements mean that KeyIdentifier attributes will all be 8 byte SHA-1 Hashes.
3398 -- existingSubjectKeyIdentifier shall be set accordingly based on the contents of the Trust Anchor Cell
3399
3400 existingSubjectKeyIdentifier      OCTET STRING
3401 }
3402
3403 TrustAnchorCellIdentifier ::=      SEQUENCE
3404 {
3405 -- Which Remote Party Role does this Cell relate to?
3406
3407 trustAnchorCellRemotePartyRole    RemotePartyRole,
3408
3409 -- To what cryptographic use can the Public Key in this Cell be put? Some Remote Party Roles
3410 -- (e.g. supplier) can have more than one Public Key on a Device and each one would only have
3411 -- a single cryptographic use.
3412
3413 trustAnchorCellKeyUsage           KeyUsage,
3414
3415 -- trustAnchorCellUsage is to allow for multiple Public Keys of the same keyUsage for the same Remote
3416 -- Party Role. This may be absent except where used to refer to the Supplier Key
3417 -- Agreement Key used solely in relation to validating Supplier generated MACs on Prepayment Top Up transactions
3418
3419 trustAnchorCellUsage           CellUsage DEFAULT management
3420 }
3421
3422 CellUsage ::=                      INTEGER {management(0), prePaymentTopUp(1)}
3423
3424 RemotePartyRole ::=                INTEGER
```

```
3425 {
3426   -- Define the full set of Remote Party Roles in relation to which a Device may need to undertake
3427   -- processing. Note that most Devices will only support processing in relation to a subset of these.
3428
3429   root                      (0),
3430   recovery                  (1),
3431   supplier                 (2),
3432   networkOperator           (3),
3433   accessControlBroker      (4),
3434   transitionalCoS          (5),
3435   wanProvider               (6),
3436   issuingAuthority          (7),    -- Devices will receive such Certificates but they do not
3437                           -- need to store them over an extended period
3438
3439
3440
3441   -- The 'other' RemotePartyRole is for a party whose role does not allow it to invoke any Device function apart from
3442   -- UpdateSecurityCredentials. This is to allow for Device functionality to be locked out of usage until a valid
3443   -- Remote Party can be identified e.g. where roles cannot be fixed until a Device is bought in to operation
3444   other                      (127)
3445
3446 }
3447
3448   -- KeyUsage is only repeated here for ease of reference. It is defined in RFC 5912
3449
3450   KeyUsage ::=                   BIT STRING
3451   {
3452     -- Define valid uses of Public Keys.
3453
3454     digitalSignature            (0),
3455     contentCommitment          (1),    -- not valid for GBCS compliant transactions
3456     keyEncipherment            (2),    -- not valid for GBCS compliant transactions
3457     dataEncipherment           (3),
3458     keyAgreement               (4),
3459     keyCertSign                (5),
3460     cRLSign                    (6),
3461     encipherOnly               (7),
3462     decipherOnly                (8)    -- not valid for GBCS compliant transactions
3463   }
3464
3465   -- The GBCS only allows for a constrained set of Trust Anchor Cell operations and so the list of possible outcomes
3466   -- is more limited than in IETF RFC 5934. The list below is that more constrained subset
3467
```

```

3468 StatusCode ::= ENUMERATED {
3469   success          (0),
3470
3471   -- trustAnchorNotFound indicates that details of a trust anchor were requested, but the referenced trust anchor
3472   -- is not represented on the Device
3473   trustAnchorNotFound      (25),
3474
3475   other             (127)
3476
3477 END
3478
3479
3480
3481

```

13.2.4 Provide Security Credential Details from a Device – Processing Steps

This Section 13.2.4 lays out the requirements relating to the construction, protection and Authentication of the Provide Security Credentials Command, and the construction, protection and Authentication of the corresponding Response.

13.2.4.1 Command Construction

The Remote Party constructing the Command shall populate Grouping Header according to the requirements of Section 7.2.6.

@ProvideSecurityCredentialDetails.Command shall have the structure defined in Section 13.2.3.3, and the Remote Party constructing the Command shall populate with values according to Table 13.2.4.1.

The Remote Party constructing the Command shall populate Command Length once it has fully populated @ProvideSecurityCredentialDetails.Command, based on the length of the octet string so constructed.

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
@ProvideSecurityCredentialDetails.Command ::=	SEQUENCE			
authorisingRemotePartyTACellIdentifier	SEQUENCE		Mandatory	This structure identifies which Public Key on the Device is to be used in checking the Command's cryptographic protection . The key is identified by way of Trust Anchor Cell and so the nature of the check, by way of the

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
				KeyUsage parameter, is also identified
trustAnchorCellRemotePartyRole	INTEGER	recovery (1) , supplier (2) , networkOperator (3) , accessControlBroker (4) , transitionalCoS (5) , wanProvider (6)	Mandatory if authorisingRemotePartyTACellIdentifier present	The role of the Party applying the Command's cryptographic protection
trustAnchorCellKeyUsage	BIT STRING	digitalSignature (0) , keyAgreement (4)	Mandatory if authorisingRemotePartyTACellIdentifier present	Where the Command's cryptographic protection is a digital signature (digitalSignature) or a MAC (keyAgreement). The value shall be digitalSignature unless trustAnchorCellRemotePartyRole = accessControlBroker
trustAnchorCellUsage	INTEGER	management(0)	DEFAULT management	Must be absent or set to 'management' since the prePaymentTopUp key pair cannot be used in relation to this command
remotePartyRolesCredentialsRequired	SEQUENCE OF			
RemotePartyRole	INTEGER	root (0) , recovery (1) , supplier (2) , networkOperator (3) , accessControlBroker (4) , transitionalCoS (5) , wanProvider (6)	Mandatory to have at least one	List the Remote Party Role(s) for which credential details are required

Table 13.2.4.1: Attribute values for Provide Security Credentials Command

3490 **13.2.4.2 Command Cryptographic Protection**

3491 If the Access Control Broker is undertaking this step to apply a MAC, then the Access Control Broker shall undertake the steps in Section
 3492 13.2.4.2.1 otherwise:

- 3493 • the Remote Party originating the Command shall generate a Signature for the Command and set KRP Signature accordingly;
- 3494 • the Signature, for incorporation in the Command, shall only be generated once all fields of the Grouping Header ||
 3495 @ProvideSecurityCredentialDetails.Command are populated as per the requirements for the Command Construction stage; and
- 3496 • the Remote Party shall use its Private Digital Signing Key to generate the Signature.

3497 **13.2.4.2.1 Access Control Broker MAC**

3498 If the Access Control Broker is undertaking this step to apply a MAC, then the Access Control Broker shall calculate a MAC using the
 3499 parameters in Table 13.2.4.2.1 and set ACB-SMD MAC to the value so calculated.

Input Parameter	Value	Note
To calculate the Shared Secret ('Z') input to the KDF:		
Private Key Agreement Key	Access Control Broker's	
Public Key Agreement Key	Device's	As identified by the Business Target ID in Message Identifier
The other input to the KDF ('OtherInfo') shall be calculated according to the requirements of Section 4.3.3.3.		
As input to the GMAC function, the IV shall be constructed according to the requirements of Section 4.3.3.4, the Plaintext shall be empty and:		
Additional Authenticated Data shall be the concatenation:	0x11 Grouping Header @ProvideSecurityCredentialDetails.Command 0x00	

3500 Table 13.2.4.2.1: Calculation of Access Control Broker MAC for Provide Security Credentials command

3501 **13.2.4.3 Command Authenticity and Integrity Verification**

3502 The Device shall undertake processing according to the requirements of this section before undertaking any other processing of the Command.
 3503 The checks should be carried out in the order specified. The Device shall cease checking at the point that any one check fails.
 3504 The checks required are shown in Table 13.2.4.3.

Check Number	Criteria that shall be tested by the Device	How the Device shall test the Criteria
1.1	The Message is for the Device	The value in the Business Target ID field of the Message Identifier part of the Command instance must be equal to the Device's Entity Identifier
1.2	The Message Code is for Provide Security Credentials	The value in the Message Code field of the Command instance must be equal to 0x0008
2.1	The Command was protected cryptographically using the Private Key corresponding to the Remote Party Public Key held in the Trust Anchor Cell identified by authorisingRemotePartyTACellIdentifier	As specified in Section 13.2.4.3.1

3505 Table 13.2.4.3: Provide Security Credentials Command authenticity and integrity verification

3506 Should any of the checks detailed in this Section 13.2.4.3 fail then the Device shall:

- 3507 • generate an entry in the Security Log recording failed Authentication;
- 3508 • discard the Command without execution and without sending a Response; and
- 3509 • send an Alert notifying the failed Authentication, constructed as specified in Section 6.2.4.2, populated with the relevant Alert Code from
3510 Section 16, to the Known Remote Party identified by the Security Credentials it holds in the {supplier, management,
3511 digitalSignature} Trust Anchor Cell.

3512 Where all of the checks detailed in this Section 13.2.4.3 succeed the Device shall process the Command and produce a Response.

3513 *13.2.4.3.1 Command Authenticity and Integrity Verification*

3514 The Device shall undertake the following checks until either all are successful or one has failed.

- 3515 1. If trustAnchorCellUsage is present it has a value of management else this test shall fail.
- 3516 2. If trustAnchorCellKeyUsage = keyAgreement then
 ((trustAnchorCellRemotePartyRole = accessControlBroker) and (the MAC calculated by the Device according to Table
3517 13.2.4.3.1 equates to ACB-SMD MAC))
 else
 ((trustAnchorCellKeyUsage = digitalSignature) and (the Device shall use the Public Key in the Trust Anchor Cell
3518 identified by authorisingRemotePartyTACellIdentifier to verify that KRP Signature is the Digital Signature across
3519 Grouping Header || @ProvideSecurityCredentialDetails.Command))

3523 else
 3524 3. This test shall fail.

Input Parameter	Value	Note
To calculate the Shared Secret ('Z') input to the KDF:		
Private Key Agreement Key	Device's	
Public Key Agreement Key	Access Control Broker's	As held by the Device in Trust Anchor Cell {accessControlBroker, keyAgreement, management}
The other input to the KDF ('OtherInfo') shall be calculated according to the requirements of Section 4.3.3.3.		
As input to the GMAC function, the IV shall be constructed according to the requirements of Section 4.3.3.4, the Plaintext shall be empty and:		
Additional Authenticated Data shall be the concatenation:	0x11 Grouping Header @ProvideSecurityCredentialDetails.Command 0x00	

3525 Table 13.2.4.3.1: Calculation of MAC for Provide Security Credential Details Command

13.2.4.4 Response Construction

3526 The Device shall populate Grouping Header according to the requirements of Section 7.2.6.

3527 The @ProvideSecurityCredentialDetails.Response shall have the structure defined in Section 13.2.3.3, and the Device shall
 3529 populate with values according to Table 13.2.4.4.

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
@ProvideSecurityCredentialDetails.Response ::=	SEQUENCE OF			
SEQUENCE				
remotePartyRole	INTEGER	root (0), recovery (1),	Mandatory if SEQUENCE is present	The role to which the credentials in this SEQUENCE relate

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
		supplier (2) , networkOperator (3) , accessControlBroker (4) , transitionalCoS (5) , wanProvider (6) ,		
statusCode	ENUMERATED	success (0) , trustAnchorNotFound (25) , other (127)	Mandatory if SEQUENCE is present	Whether the Device can supply the details
currentSeqNumber	INTEGER	The corresponding Counter value	Present if statusCode=0	The Protection Against Replay number held by the Device for this role's use of the Update Security Credentials Command
trustAnchorCellsDetails	SEQUENCE OF		At least one in the SEQUENCE OF must be present if statusCode=0	
SEQUENCE				
trustAnchorCellKeyUsage	BIT STRING	digitalSignature (0) , keyAgreement (4) , keyCertSign (5)	Mandatory if SEQUENCE is present	To what use can the public key in this Cell be put
trustAnchorCellUsage	INTEGER	prePaymentTopUp(1)	DEFAULT management	Only needs to be present for the {supplier, keyAgreement,

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
			(0)	prePaymentTopUp} Cell
existingSubjectUniqueID	OCTET STRING	Entity Identifier in this Cell	Mandatory if SEQUENCE is present	See Section 12.4
existingSubjectKeyIdentifier	OCTET STRING	Key Identifier of the key in this Cell	Mandatory if SEQUENCE is present	

3530 Table 13.2.4.4: Attribute values for Provide Security Credentials Response

3531 **13.2.4.5 Response Cryptographic Protection**3532 If the Command that triggered this Response was protected by a MAC then the Device shall calculate a MAC using the parameters in Table
3533 13.2.4.5 and set SMD-KRP MAC to the value so calculated.

Input Parameter	Value	Note
To calculate the Shared Secret ('Z') input to the KDF:		
Private Key Agreement Key	Device's	
Public Key Agreement Key	Access Control Broker's	As held by the Device in {accessControlBroker, keyAgreement, Management}
The other input to the KDF ('OtherInfo') shall be calculated according to the requirements of Section 4.3.3.3.		
As input to the GMAC function, the IV shall be constructed according to the requirements of Section 4.3.3.4, the Plaintext shall be empty and:		
Additional Authenticated Data shall be the concatenation:	0x11 Grouping Header @ProvideSecurityCredentialDetails.Response 0x00	

3534 Table 13.2.4.5: Calculation of MAC for Provide Security Credential Details Response

3535 Otherwise:

- 3536 • the Device creating the Response shall generate a Signature for the Response and set SMD Signature to the value calculated;
 3537 • the Signature, for incorporation in the Response, shall only be generated once all fields of the Grouping Header || Length ||
 3538 @ProvideSecurityCredentialDetails.Response are populated, as per requirements for the Response Construction stage; and
 3539 • the Device shall use its Private Digital Signing Key to generate the Signature.

3540 ***13.2.4.6 Response Recipient Cryptographic Verification***

3541 If the Response contains a MAC, the Access Control Broker can verify that MAC by calculating a MAC according to the parameters in Table
 3542 13.2.4.6 and checking that the MAC so calculated equates to that in the Response.

Input Parameter	Value	Note
To calculate the Shared Secret ('Z') input to the KDF:		
Private Key Agreement Key	Access Control Broker's	
Public Key Agreement Key	Device's	As identified by Business Originator ID in the Message Identifier
The other input to the KDF ('OtherInfo') shall be calculated according to the requirements of Section 4.3.3.3.		
As input to the GMAC function, the IV shall be constructed according to the requirements of Section 4.3.3.4, the Plaintext shall be empty and:		
Additional Authenticated Data shall be the concatenation:	0x11 Grouping Header @ProvideSecurityCredentialDetails.Response 0x00	

3543 Table 13.2.4.6: Calculation of MAC for Provide Security Credential Details Response Verification

3544 ***13.3 CS02b Update Security Credentials Command, Response and Alert***

3545 ***13.3.1 Description***

3546 This Section 13.3 covers the creation, validation and processing of:

- 3547 • Update Security Credentials Commands;
 3548 • Responses to such Commands; and
 3549 • Alerts resulting from the future dated execution of such Commands.

3550 The Update Security Credentials Command shall be:

- 3551 • used solely to replace Remote Party Security Credentials held in Trust Anchor Cells on Devices;
- 3552 • supported by any Device that can process Remote Party Messages; and
- 3553 • the only Command that Devices are capable of accepting for replacement of Remote Party Security Credentials, once the tamper seal is applied to the Device.
- 3554

13.3.2 Use Case Cross References

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Command and Response / Alert
Message Type Category	Variant Message and is Critical
Capable of future dated invocation?	Yes (but see constraint in Table 13.3.5.1, check 1.3)
Protection Against Replay Required?	The Protection Against Replay mechanisms for Update Security Credentials are specified in Section 13.3. The Protection Against Replay mechanisms of other sections of the GBCS do not apply.
SEC User Gateway Services Schedule (Service Request) Reference	6.15, 6.23, 8.5, 6.21
Valid Target Device(s)	ESME / GSME / GPF / CHF / HCALCS / PPMID
Valid Business Originator role(s) for Command invocation (and so, for DLMS COSEM Commands, which Application Association is to be used for delivery of the Command to the Device) [Remote Party Messages Only]	Supplier Network Operator Access Control Broker Transitional Change of Supplier WAN Provider Recovery
Valid Response Recipient role(s) (only for Messages Authorised by the Access Control Broker on behalf of parties not known to the Device) [Remote Party Messages Only]	N/A
Valid initiating Device type(s) [HAN Only Messages]	N/A
Protocol	ASN.1

3556 Table 13.3.2: Use Case Cross References for Update Security Credentials Command

3557 **13.3.3 Command, Response and Alert Structure**

3558 **13.3.3.1 The Update Security Credentials Command**

3559 This Section 13.3.3.1 summarises the structure of the Update Security Credentials Command, which depends on
3560 credentialsReplacementMode and the deviceType of the Device.

3561 If credentialsReplacementMode = anyByContingency or anyExceptAbnormalRootByRecovery then an Update Security Credential
3562 Details Command shall be the concatenation:

3563 Grouping Header || @UpdateSecurityCredentials.CommandPayload || 0x40 || KRP Signature

3564 If credentialsReplacementMode = accessControlBrokerByACB and deviceType is not
3565 communicationsHubCommunicationsHubFunction then an Update Security Credentials Command shall be the concatenation:

3566 MAC Header || Grouping Header || @UpdateSecurityCredentials.CommandPayload || 0x00 || ACB-SMD MAC

3567 In all other cases, the Update Security Credentials Command shall either be the concatenation:

3568 MAC Header || Grouping Header || @UpdateSecurityCredentials.CommandPayload || 0x40 || KRP Signature|| ACB-SMD MAC

3569 In these Command structures:

- 3570 • @UpdateSecurityCredentials.CommandPayload shall be an octet string containing the DER encoding of the populated ASN.1
3571 structure (as laid out in Section 13.3.5.11);
- 3572 • Grouping Header shall be constructed as specified in Section 7.2.7 with Business Originator ID being the Entity Identifier of the Known
3573 Remote Party which generated KRP Signature, and with Business Originator Counter being that of the same Known Remote Party;
- 3574 • KRP Signature shall be generated as specified in Section 6.3.3;
- 3575 • ACB Grouping Header shall be constructed as specified in Section 7.2.7 with Business Originator ID being the Entity Identifier of the
3576 Access Control Broker and Business Originator Counter being that of the Access Control Broker;
- 3577 • MAC Header shall be constructed as specified in Section 7.2.5; and
- 3578 • ACB-SMD MAC shall be calculated as specified in Section 6.2.3.

3579 **13.3.3.2 The Update Security Credentials Response**

3580 An Update Security Credentials Response shall be the concatenation:

3581 **Grouping Header || @UpdateSecurityCredentials.ResponsePayload || 0x40 || SMD Signature**

3582 where:

- 3583 • @UpdateSecurityCredentials.ResponsePayload shall be an octet string containing the DER encoding of the populated ASN.1
3584 structure (as laid out in Section 13.3.5.11);
3585 • Grouping Header in the Response shall be constructed as specified in Section 7.2.7 with Business Target ID being the Entity Identifier
3586 specified in the corresponding Command's Grouping Header; and
3587 • SMD Signature shall be generated as specified in Section 6.3.5.

3588 ***13.3.3.3 The Update Security Credentials Alert***

3589 **An Update Security Credentials Alert shall be the concatenation:**

3590 **Grouping Header || @UpdateSecurityCredentials.AlertPayload || 0x40 || SMD Signature**

3591 where:

- 3592 • @UpdateSecurityCredentials.AlertPayload shall be an octet string containing the DER encoding of the populated ASN.1
3593 structure (as laid out in Section 13.3.5.11);
3594 • Grouping Header in the Alert shall be constructed as specified in Section 7.2.7 with Business Target ID being the Entity Identifier specified
3595 in the corresponding Command's Grouping Header;
3596 • the Message Code being 0x00CB; and
3597 • SMD Signature shall be generated as specified in Section 6.3.5.

3598 ***13.3.3.4 The Update Security Credentials Command, Response and Alert - informative***

3599 **The @UpdateSecurityCredentials.CommandPayload structure has four parts:**

- 3600 • **authorisingRemotePartyControl**: which includes details of what kind of credential replacement this Command is, which Remote
3601 Parties are authorising it and information to support Protection Against Replay protections;
3602 • **replacements**: which is a list of new Certificates the Device is to store details from, along with which Trust Anchor Cell each set of details
3603 is to be stored in on the Device;
3604 • **certificationPathCertificates**: which is a list of Certification Authority Certificates the Device will need to use in checking that the
3605 replacement Certificates were properly issued; and

- 3606 • `executionDateTime`: which, if present, specifies the date-time at which the certificates in the `CommandPayload` are to be used to
 3607 replace the credentials currently in use on the Device. If this field is not present, the Command shall be executed immediately. If this field
 3608 has the value equivalent to 'never' (which is '99991231235959Z') the certificate replacement will never happen. This is to allow
 3609 cancellation of future dated Commands. Note that future dating is not supported where certificates are being replaced in exception
 3610 conditions.

3611 The `@UpdateSecurityCredentials.Response` structure contains, for immediate execution commands, a list detailing the success of
 3612 failure of each of the replacements, including details of the parties affected. For future dated commands,
 3613 `@UpdateSecurityCredentials.AlertPayload` structure contains the list detailing the success, or failure, of each of the replacements,
 3614 including details of the parties affected.

3615 Section 13.3.5.11 contains narrative for each of the parts of these ASN.1 structures.

3616 Section 18.2.1.2 provides an illustrative instantiation of `@UpdateSecurityCredentials.CommandPayload` and its corresponding DER
 3617 encoding.

3618 13.3.4 Updating Security Credentials on a Device – Processing Steps

3619 This section lays out the requirements for the construction, protection and Authentication of the Update Security Credentials Command Payload,
 3620 the processing required on the Device of the Command, the construction of the corresponding Response Payload and, where required, the
 3621 Alert Payload.

3622 13.3.4.1 Command Payload Construction

3623 The `@UpdateSecurityCredentials.CommandPayload` shall have the structure defined in Section 13.3.5.11, and the Remote Party
 3624 constructing the Command shall populate with values according to Table 13.3.4.1.

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
<code>@ UpdateSecurityCredentials.Command ::=</code>	SEQUENCE			
<code>authorisingRemotePartyControl</code>	SEQUENCE			This structure provides details to allow the Device to identify the Remote Party Role authorising this Command, check whether the rest of the payload is allowable

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
				and allow counters / counter caches on the Device to be reset, if the command changes the Remote Party in control
credentialsReplacementMode	INTEGER	rootBySupplier (0), rootByWanProvider (1), supplierBySupplier (2), networkOperatorByNetworkOperator (3), accessControlBrokerByACB (4), wanProviderByWanProvider (5), transCoSByTransCoS (6), supplierByTransCoS (7), anyExceptAbnormalRootByRecovery (8), anyByContingency (9)	Mandatory	Specify the replacement mode so that the Device can check that the Remote Party Role authorising the command is allowed to authorise this type of replacement(s) and that all replacements in the payload are allowed within this replacement mode. The structure of the label is <i>kindOfCertificate(s)BeingReplacedBypartydoingtherreplacement</i> . For example, rootBySupplier is where a new root Certificate is being provided to the Device by its Supplier
plaintextSymmetricKey	[0] IMPLICIT OCTET STRING	The symmetric key that will decrypt the encrypted Contingency Key held on the Device	OPTIONAL	Only to be present if the Contingency Key arrangements are being used (so if credentialsReplacementMode = anyByContingency). The contents provide the symmetric

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
				key to decrypt the Contingency Public Key in the (<code>root</code> , <code>digitalSignature</code> , <code>management</code>) Trust Anchor Cell
<code>applyTimeBasedCPVChecks</code>	[1] IMPLICIT INTEGER	disapply(1)	DEFAULT apply	Only to be present if the Remote Party sending the Command is instructing the Device not to apply time based checks as part of Certification Path Validation. This should only be in exceptional circumstances (e.g. root credentials on the Device have expired without replacement for unforeseen reasons)
<code>authorisingRemotePartyTACellIdentifier</code>	[2] IMPLICIT SEQUENCE		OPTIONAL	This structure identifies which Public Key on the Device is to be used in verifying KRP Signature. The key is identified by way of Trust Anchor Cell and so the nature of the check, by way of the KeyUsage parameter, is also identified. <code>'authorisingRemotePartyTACellIdentifier'</code> can only be omitted when the Access Control Broker is changing its own Key Agreement credentials
<code>trustAnchorCellRemotePartyRole</code>	INTEGER	root (0), recovery (1), supplier (2), networkOperator (3), accessControlBroker	Mandatory if <code>authorisingRemotePartyTACellIdentifier</code> present	The role of the Party applying KRP Signature.. Note that where root is used, this refers only to the encrypted Contingency key in the root TA Cell, so is only valid if <code>credentialsReplacementMode = anyByContingency</code> and

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
		er (4), transitionalCoS (5), wanProvider (6)		plaintextSymmetricKey is populated with the symmetric key required to decrypt that public key
trustAnchorCellKeyUsage	BIT STRING	digitalSignature (0)	Mandatory if authorisingRemotePartyTA CellIdentifier present	KRP Signature is a digital signature
trustAnchorCellUsage	INTEGER	management (0)	DEFAULT management	Must be absent or set to 'management' since the prePaymentTopUp key pair cannot be used in relation to this Command
authorisingRemotePartySeqNumber	[3] IMPLICIT INTEGER	Originator Counter of Remote Party authorising the Command	Mandatory	Specify the Originator Counter for the Remote Party applying KRP Signature, or (for the Access Control Broker changing its credentials) the Access Control Broker's Originator Counter
newRemotePartyFloorSeqNumber	[4] IMPLICIT INTEGER	Originator Counter of Remote Party who will have control of this Remote Party Role if the update is successful	OPTIONAL	If the Command is to effect a change of control, then newRemotePartyFloorSeqNumber should be included and will be the value used to prevent replay of Update Security Credentials Commands, and other Commands, for the new controlling Remote Party
newRemotePartySpecialistFloorSeqNumber	[5] IMPLICIT		OPTIONAL	Some Commands on the Device may use a different Originator

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
	SEQUENCE OF			Counter sequence for Protection Against Replay. The only example is the Prepayment Top Up Command on ESME and GSME. The SpecialistSeqNumber structure allows such Counters to also be reset on change of control. Should only be present if this Command changes supplier credentials and the new supplier uses different counters for its Prepayment Top Ups than it does for other Commands
SEQUENCE				
seqNumberUsage	INTEGER	prepaymentTopUp (0)	Mandatory if newRemotePartySpecialistFloorSeqNumber present	Specify the usage of the SeqNumber
seqNumber	INTEGER	Relevant Originator Counter	OPTIONAL	Specify the associated SeqNumber
otherRemotePartySeqNumberChanges	[6] IMPLICIT SEQUENCE OF		OPTIONAL	In some cases, one party acting in one Remote Party Role may be replacing certificates for a different Remote Party Role (e.g. transitionalCoS changing Supplier Credentials). In such cases, sequence counters need also to be reset for that other

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
				Remote Party Role
SEQUENCE				
otherRemotePartyRole	INTEGER	supplier (2) , networkOperator (3) , accessControlBroker (4) , transitionalCoS (5) , wanProvider (6) ,	Mandatory if otherRemotePartySeqNumberChanges present	The Remote Party Role of the party whose credentials are being placed on the Device but which didn't authorise the command directly. Note that this is not valid for root or recovery
otherRemotePartyFloorSeqNumber	INTEGER	Relevant Originator Counter	Mandatory if otherRemotePartySeqNumberChanges present	Specify the associated SeqNumber
otherRemotePartySpecialistFloorSeqNumber	SEQUENCE OF		OPTIONAL	Should only be present if otherRemotePartyRole = supplier, and that new supplier uses different counters to prevent replay on Prepayment Top Up
SEQUENCE				
seqNumberUsage	INTEGER	prepaymentTopUp (0)	Mandatory if newRemotePartySpecialistFloorSeqNumber present	Specify the usage of the SeqNumber
seqNumber	INTEGER	Relevant Originator Counter	OPTIONAL	Specify the associated SeqNumber

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
replacements	SEQUENCE OF			Provide a list of the replacements. Each replacement contains a new end entity' Certificate and the identity of the Trust Anchor Cell which is to have its contents replaced using that Certificate.
SEQUENCE			At least one SEQUENCE must be present	One structure is required for each Trust Anchor Cell that is to be updated
replacementCertificate	Certificate	End entity Certificate	Mandatory if SEQUENCE is present	Provide the new end entity certificate
targetTrustAnchorCell	SEQUENCE			Specify where it is to go (specifically which Trust Anchor Cell is to have its details replaced using the new end entity certificate)
trustAnchorCellRemotePartyRole	INTEGER	root (0) , recovery (1) , supplier (2) , networkOperator (3) , accessControlBroker (4) , transitionalCoS (5) , wanProvider (6)	Mandatory if SEQUENCE is present	To which Remote Party Role does the Trust Anchor Cell relate

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
trustAnchorCellKeyUsage	BIT STRING	{digitalSignature(0), keyAgreement(4), keyCertSign(5)} ,	Mandatory if SEQUENCE is present	To what use can the public key in this Cell be put
trustAnchorCellUsage	INTEGER	prePaymentTopUp(1) }	DEFAULT management	Should be absent unless: <ul style="list-style-type: none"> the deviceType is eSME or gSME; and the supplier operating the Device has chosen to use a separate key agreement Key Pair in relation to prepayment top ups to the Device and this is a replacement of the corresponding certificate
certificationPathCertificates	SEQUENCE OF Certificate	The list of certificates needed for Certification Path Validation	At least one Certificate must be present since root will never directly sign any end entity certificate	Provide the certificates needed to undertake Certification Path Validation of the new end entity certificate against the root public key held on the Device. The number of these may be less than the number of replacement certificates (e.g. a supplier may replace all of its certificates but may only need to supply one Certification Authority Certificate to link them all back to root)

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
executionDateTime	GeneralizedTime	The date-time at which the replacements are to be used in updating the Device's Security Credentials	OPTIONAL	This field may only be present if credentialsReplacementMode is either supplierBySupplier or supplierByTransCoS

3625 Table 13.3.4.1: Attribute values for Update Security Credentials Command

3626 **13.3.4.2 Command Authenticity and Integrity Verification**

3627 The Device shall undertake processing according to the requirements of Section 13.3.5.1.

3628 Should any of the checks detailed in Section 13.3.5.1 fail then the Device shall:

- 3629 • generate an entry in the Security Log recording failed Authentication;
- 3630 • discard the Command without execution and without sending a Response; and
- 3631 • send an Alert notifying the failed Authentication, constructed as specified in Section 6.2.4.2 of the GBCS, populated with the relevant Alert Code according to Section 16, to the Known Remote Party identified by:

3632

- 3633 ○ the Trust Anchor Cell {supplier, digitalSignature, management} if the Device's deviceType is not communicationsHubFunction; or

3634

- 3635 ○ the Trust Anchor Cell {wanProvider, digitalSignature, management} if the Device's deviceType is communicationsHubFunction.

3637 Where all of the checks detailed in Section 13.3.5.1 succeed the Device shall process the Command and produce a Response.

3638 **13.3.4.3 Command Processing**

3639 Before undertaking any further processing, the Device shall update Highest Prior Sequence Number to the value of authorisingRemotePartySeqNumber.

3641 If executionDateTime is present then the Device shall:

- 3642 • record against the `remotePartyRole` (as specified in `authorisingRemotePartyControl`), `authorisingRemotePartyControl`,
3643 replacements; and `executionDateTime`;

- 3644 • construct a Response where `executionOutcome` is not present according to the requirements of Section 13.3.4.4; and

- 3645 • at the date-time specified in `executionDateTime`, undertake the processing of Section 13.3.4.3.1 then construct an Alert according to
3646 the requirements of Section 13.3.4.5.

3647 If `executionDateTime` is not present then the Device shall:

- 3648 • undertake the processing of Section 13.3.4.3.1; and

- 3649 • construct a Response where `executionOutcome` is present according to the requirements of Section 13.3.4.4.

3650 **13.3.4.3.1 replacements Processing**

3651 For each of the `targetTrustAnchorCell` in `replacements`, the Device shall:

- 3652 • record the `entityIdentifier` and `subjectKeyIdentifier` currently held in that `targetTrustAnchorCell`;

- 3653 • attempt to replace the contents of that `targetTrustAnchorCell` using the corresponding certificate in
3654 `TrustAnchorReplacement`; and

- 3655 • if the contents of the replacement are successfully applied, undertake the processing required by Section 13.3.5.10 in relation to the
3656 `RemotePartyRole` for that `targetTrustAnchorCell`.

3657 **13.3.4.4 Response Construction**

3658 The `@UpdateSecurityCredentials.ResponsePayload` shall have the structure defined in Section 13.3.5.11, and the Device shall
3659 populate the `executionOutcome`, where present with values according to Table 13.3.4.4.

3660 **13.3.4.5 Alert Construction**

3661 The `@UpdateSecurityCredentials.AlertPayload` shall have the structure defined in Section 13.3.4, and the Device shall populate the
3662 `executionOutcome`, with values according to Table 13.3.4.6.

3663

13.3.4.6 executionOutcome Construction

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
executionOutcome	SEQUENCE			
authorisingRemotePartySeqNumber	INTEGER	Originator Counter of Remote Party authorising the Command, as specified in the corresponding Command	Mandatory	This is to allow the Alert to be linked to the Command that caused execution
credentialsReplacementMode	INTEGER	rootBySupplier (0) , rootByWanProvider (1) , supplierBySupplier (2) , networkOperatorByNetworkOperator (3) , accessControlBrokerByACB (4) , wanProviderByWanProvider (5) , transCoSByTransCoS (6) , supplierByTransCoS (7) , anyExceptAbnormalRootByRecovery (8) , anyByContingency (9) } ,	Mandatory	Provide details of the corresponding Command that are not in the standard GBCS message header. Specifically the mode in which the Command was invoked

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
remotePartySeqNumberChanges	SEQUENCE OF		OPTIONAL	The resulting changes to any replay counters held on the Device
SEQUENCE				
otherRemotePartyRole	INTEGER	root (0) , recovery (1) , supplier (2) , networkOperator (3) , accessControlBroker (4) , transitionalCoS (5) , wanProvider (6) ,	Mandatory if SEQUENCE is present	The role which has had its counter values changed on the Device
otherRemotePartyFloorSeqNumber	INTEGER	The corresponding Counter value	Mandatory if SEQUENCE is present	
newRemotePartySpecialistFloorSeqNumber	SEQUENCE OF		OPTIONAL	Only present where Remote Party Role is supplier
SEQUENCE				
seqNumberUsage	INTEGER	{prepaymentTopUp (0)} ,	Mandatory if newRemotePartySpecialistFloorSeqNumber present	Specify the usage of the SeqNumber
seqNumber	INTEGER		Mandatory if newRemotePartySpecialistFloorSeqNumber present	Specify the associated SeqNumber

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
			stFloorSeqNumber present	
replacementOutcomes	SEQUENCE OF		One per replacement in the corresponding Command so at least one	For each replacement in the Command, detail the outcome and impacted parties
SEQUENCE				
affectedTrustAnchorCell	SEQUENCE		Mandatory if SEQUENCE is present	Specify which Trust Anchor Cell was the target of this replacement
trustAnchorCellRemotePartyRole	INTEGER	root (0) , recovery (1) , supplier (2) , networkOperator (3) , accessControlBroker (4) , transitionalCoS (5) , wanProvider (6)	Mandatory if SEQUENCE is present	Specify the Remote Party Role to which the Trust Anchor Cell relates
trustAnchorCellKeyUsage	BIT STRING	digitalSignature (0) , keyAgreement (4) , keyCertSign (5)	Mandatory if SEQUENCE is present	To what use can the public key in this Cell be put
trustAnchorCellUsage	INTEGER	{management(0) ,	DEFAULT management	Absent unless:

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
		prePaymentTopUp(1) }		<ul style="list-style-type: none"> the deviceType is eSME or gSME; and the Supplier operating the Device has chosen to use a separate key agreement Key Pair in relation to prepayment top ups to the Device and this is a replacement of the corresponding certificate
statusCode	ENUMERATED	success (0) , badCertificate (5) , noTrustAnchor (10) , insufficientMemory (17) , contingencyPublicKeyDecrypt (22) , trustAnchorNotFound (25) , resourcesBusy (30) , other (127)	Mandatory if SEQUENCE is present	Whether the replacement to this Cell was successful or, if it failed, why it failed
existingSubjectUniqueID	OCTET STRING		Mandatory if SEQUENCE is present	The 64 bit Entity Identifier of the Remote Party whose credentials were in this Cell prior to receipt of the corresponding Command
existingSubjectKeyIdentifier	OCTET STRING		Mandatory if SEQUENCE is present	For the public key in this Cell prior to receipt of the corresponding Command
replacingSubjectUniqueID	OCTET STRING		Mandatory if SEQUENCE is present	The 64 bit Entity Identifier of the Remote Party whose credentials were to be placed in this Cell

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
replacingSubjectKeyIdentifier	OCTET STRING		Mandatory if SEQUENCE is present	For the public key which was to be placed in this Cell

Table 13.3.4.4: Attribute values for executionOutcome

3664

13.3.5 Common Requirements

3665

13.3.5.1 Update Security Credentials Command Verification

3666

3667

3668

The Device shall undertake the checks set out in this Section 13.3.5.1 before undertaking any other processing of the Command. The checks should be carried out in the order specified. Checking shall cease at the point that any one check fails. The checks required are shown in Table 13.3.5.1.

Check Number	Criteria that must be tested by the Device	How the Device shall test the Criteria
1.1	The Message is for the Device	The value of the Business Target ID in the Grouping Header in Command instance must be equal to the Device's Entity Identifier
1.2	The Message Code is for Update Security Credentials	The value in the Message Code field of the Grouping Header must be equal to the value specified in Table 13.3.5.2 for the CredentialsReplacementMode specified in CommandPayload.
1.3	If executionDateTime is present the Command is to replace Supplier Security Credentials.	If executionDateTime is present then credentialsReplacementMode must either supplierBySupplier or supplierByTransCos
1.4	The Device has not already actioned this Command.	As specified in Section 13.3.5.3
2.1	The targetTrustAnchorCells all exist on a Device of this type	As specified in Section 13.3.5.4
2.2	The credentialsReplacementMode is one that can be Authorised by	As specified in Section 13.3.5.5

Check Number	Criteria that must be tested by the Device	How the Device shall test the Criteria
	the Remote Party / Parties authorising the Command	
2.2	The replacements specified are all allowed in this credentialsReplacementMode .	As specified in Section 13.3.5.6
2.3	The keyUsage in each of the replacement certificates provided is consistent with the target Trust Anchor Cells identified in replacements	As specified in Section 13.3.5.7
3.1	The Cryptographic Protections are valid	As specified in Section 13.3.5.8

3669 Table 13.3.5.1: Update Security Credentials Command authenticity and integrity verification

3670 **13.3.5.2 Message Code Validation**

CredentialsReplacementMode	Message Code
rootBySupplier	0x0100
rootByWanProvider	0x0101
supplierBySupplier	0x0102
networkOperatorByNetworkOperator	0x0103
accessControlBrokerByACB	0x0104
wanProviderByWanProvider	0x0105
transCoSByTransCoS	0x0106
supplierByTransCoS	0x0107
anyExceptAbnormalRootByRecovery	0x0108
anyByContingency	0x0109

3671 Table 13.3.5.2: Message Code validation against CredentialsReplacementMode

3672 **13.3.5.3 Preventing Replay of Commands**

3673 The Protection Against Replay mechanisms for the Update Security Credentials Command shall be that specified in this Section 13.3.5.3
3674 (which is different than that for other GBCS Commands).

3675 For each of `RemotePartyRole` from which the Device can receive a valid Updated Security Credentials Command, the Device shall allocate
3676 storage for a Highest Prior Sequence Number which shall be capable of storing a 64 bit unsigned integer and which shall initially be set to the
3677 value zero at manufacture.

3678 Before executing any Update Security Credentials Command, a Device shall confirm that, if `CredentialsReplacementMode <>`
3679 `accessControlBrokerByACB`, then

- 3680 • (`authorisingRemotePartyTACellIdentifier` is populated in the Command) and (`the authorisingRemotePartySeqNumber` is
3681 strictly numerically greater than the Highest Prior Sequence Number the Device has recorded for the `RemotePartyRole` identified in
3682 `authorisingRemotePartyTACellIdentifier`)

3683 else

- 3684 • (`the authorisingRemotePartySeqNumber` is strictly numerically greater than the Highest Prior Sequence Number the Device has
3685 recorded for the `accessControlBroker`)

3686 **13.3.5.4 Required Trust Anchor Cells by Device Type**

3687 The Trust Anchor Cells specified in Section 4.3.2.5 are those required on each Device type and so are the only valid
3688 `targetTrustAnchorCells`.

3689 The Device shall ensure that all `targetTrustAnchorCells` specified in the Command instance are valid for the type of Device it is,
3690 according to the requirements of Section 4.3.2.5. A Command containing any invalid `targetTrustAnchorCells` shall not be processed any
3691 further by the Device.

3692 **13.3.5.5 Valid `credentialsReplacementMode` by Remote Party Roles authorising the Command**

3693 A Command containing a certain `credentialsReplacementMode` is only Authorised using certain types of Public-Private Key Pairs in
3694 certain ways. The Command identifies the Public Key corresponding to the Private Key used by the authorising Remote Party in the
3695 `authorisingRemotePartyTACellIdentifier` structure. Table 13.3.5.5 lists the only Authorised combinations. All other combinations
3696 represent Commands not properly Authorised and shall be rejected by a Device.

3697

3698

	authorisingRemotePartyTACellIdentifier		
	RemotePartyRole	KeyUsage	CellUsage
credentialsReplacementMode			
rootBySupplier	supplier	digitalSignature	management
rootByWanProvider	wanProvider	digitalSignature	management
supplierBySupplier	supplier	digitalSignature	management
networkOperatorByNetworkOperator	networkOperator	digitalSignature	management
accessControlBrokerByACB, only if authorisingRemotePartyTACellIdentifier is present	accessControlBroker	digitalSignature	management
wanProviderByWanProvider	wanProvider	digitalSignature	management
transCoSByTransCoS	transitionalCoS	digitalSignature	management
supplierByTransCoS	transitionalCoS	digitalSignature	management
anyExceptAbnormalRootByRecovery	recovery	digitalSignature	management
anyByContingency	root	digitalSignature	management

3699 Table 13.3.5.5: Valid credentialsReplacementMode by Remote Party Roles authorising the Command

3700 **13.3.5.6 Valid credentialsReplacementMode by the targetTrustAnchorCells specified in the Command**3701 A Command containing a certain credentialsReplacementMode can only validly replace the Security Credentials in a certain subset of
3702 Trust Anchor Cells. The Command identifies the Cells that are to have credentials replaced in each targetTrustAnchorCell within each
3703 TrustAnchorReplacement in replacements.3704 Table 13.3.5.6 below lists the only valid targetTrustAnchorCell combinations for each credentialsReplacementMode. All other
3705 combinations are invalid. A Command containing any invalid combinations shall not be processed any further by the Device.

	targetTrustAnchorCell		
	RemotePartyRole	KeyUsage	CellUsage
credentialsReplacementMode			

	targetTrustAnchorCell		
	RemotePartyRole	KeyUsage	CellUsage
credentialsReplacementMode			
rootBySupplier	root	keyCertSign	management
rootByWanProvider	root	keyCertSign	management
supplierBySupplier	supplier	any valid for GBCS	any valid for GBCS
networkOperatorByNetworkOperator	networkOperator	any valid for GBCS	any valid for GBCS
accessControlBrokerByACB	accessControlBroker	any valid for GBCS	any valid for GBCS
wanProviderByWanProvider	wanProvider	any valid for GBCS	any valid for GBCS
transCoSByTransCoS	transitionalCoS	digitalSignature	management
supplierByTransCoS	supplier	any valid for GBCS	any valid for GBCS
anyExceptAbnormalRootByRecovery	any valid for GBCS	any valid for GBCS	any valid for GBCS
anyByContingency	any valid for GBCS	any valid for GBCS	any valid for GBCS

Table 13.3.5.6: Valid credentialsReplacementMode by the targetTrustAnchorCells specified in the Command

- 3706 **13.3.5.7 Valid usage of Certificates against the targetTrustAnchorCells specified in the Command**
- 3707 [Note: Each of the 'end entity' Certificates in the Command must have the same keyUsage as the Trust Anchor Cell it is to be applied to.]
- 3708 For each instance of the TrustAnchorReplacement structure in the Command, the keyUsage in replacementCertificate shall be
 3709 equal to targetTrustAnchorCell.KeyUsage. Where this check fails for any one or more of the TrustAnchorReplacement instances,
 3710 the Command shall not be actioned by the Device.
- 3711 [Note: Save for supplier and network operator roles, each of the 'end entity' Certificates in the Command must have the same
 3712 RemotePartyRole as the Trust Anchor Cell it is to be applied to.]
- 3713 For each instance of the TrustAnchorReplacement structure in the Command where (targetTrustAnchorCell.RemotePartyRole <>
 3714 supplier) and (targetTrustAnchorCell.RemotePartyRole <> networkOperator), the RemotePartyRole in
 3715 replacementCertificate shall be equal to targetTrustAnchorCell.RemotePartyRole. Where this check fails for any one or more
 3716 of the TrustAnchorReplacement instances, the Command shall not be actioned by the Device.

3717 Notes:

- 3718 • mismatches between `RemotePartyRole` in the certificate and the target Trust Anchor Cell are admissible for supplier and
3719 networkOperator only, and are needed (see Section 4.3.2.5); and
3720 • `CellUsage` is simply a selector to allow a different Key Agreement key pair to be used for Prepayment Top Ups. However, that use of a
3721 different Key Pair is not mandated and so validation is not required; any valid supplier Key Agreement certificate may be used in this Trust
3722 Anchor Cell.

3723 **13.3.5.8 Verifying the Cryptographic Protections**

3724 In verifying Cryptographic Protections pursuant to this Section 13.3.5.8.1:

- 3725 • KRP Signature shall be verified according to the requirements in Section 4.3.2.7.2; and
3726 • ACB-SMD MAC shall be verified according to the requirements in Section 6.2.4.1.2.

3727 If `credentialsReplacementMode` = `anyByContingency` then KRP Signature shall be verified using the public key established according
3728 to the requirements of Section 13.3.5.8.1.

3729 If `credentialsReplacementMode` = `<>` `anyByContingency` then KRP Signature shall be verified using the public key identified as per
3730 Section 4.3.2.7.2.

3731 If `credentialsReplacementMode` = `accessControlBrokerByACB` and `deviceType` is not
3732 `communicationsHubCommunicationsHubFunction` then ACB-SMD MAC shall be verified as per Section 6.2.4.1.2.

3733 **13.3.5.8.1 Decrypting the contingency public key and verifying Authorising Remote Party's digital signature against that decrypted key**

3734 The Device shall decrypt the Contingency Key that it holds in Trust Anchor Cell {`root`, `digitalSignature`, `management`} by undertaking
3735 Decryption using the following parameters:

- 3736 • setting Ciphertext to be encrypted value of the Contingency Key;
3737 • setting Additional Authenticated Data to be `0x31`;
3738 • setting the Initialization Vector to be `0xFFFFFFFF0000000000000000`; and
3739 • setting the shared symmetric key to be the value in `plaintextSymmetricKey`.

3740 Where Decryption is successful, the Device shall use the Plaintext produced as the Public Key to verify KRP Signature according to the
3741 requirements at Section 6.3.4.

3742 The Contingency Key shall have been Encrypted accordingly.

3743 **13.3.5.9 Verifying the authenticity of replacement certificates**

3744 The Device shall first apply the requirements of Section 12.6 (Device processing of Certificates). If any of those checks fail, the Section
3745 13.3.5.9.1 check fails.

3746 Where Certification Path Validation is required by this Section 13.3.5.9, the application of time based checks shall be determined as follows:

- 3747 • if, in the Command, applyTimeBasedCPVChecks = disapply then time based checks shall NOT be applied by the Device;
- 3748 • otherwise time based checks shall be applied or not applied in line with the requirements of Section 4.3.2.8.

3749 If (credentialsReplacementMode <> anyByContingency) and (replacements does NOT include a targetTrustAnchorCell of
3750 {root, keyCertSign}) then the Device shall, for each replacementCertificate in replacements, undertake Certification Path
3751 Validation according to the requirements of Section 4.3.2.8.

3752 If (credentialsReplacementMode <> anyByContingency) and (replacements does include a targetTrustAnchorCell of {root,
3753 keyCertSign}) then the Device shall first undertake the checks at Section 13.3.5.9.1 in relation to the root Certificates and then shall, for
3754 each of the other replacementCertificate in replacements, undertake Certification Path Validation according to the requirements of
3755 Section 4.3.2.8.

3756 If (credentialsReplacementMode = anyByContingency) and (replacements does include a targetTrustAnchorCell of {root,
3757 keyCertSign}) then the Device shall, for each of the other replacementCertificate in replacements, undertake Certification Path
3758 Validation according to the requirements of Section 4.3.2.8. In so doing the Device shall use the details from the replacementCertificate
3759 in replacements specified for updating {root, keyCertSign} as the root for Certification Path Validation.

3760 **13.3.5.9.1 Validation of new root Certificate against current root Security Credentials**

3761 The Device shall:

- 3762 • identify the Certificate in replacements that corresponds to the targetTrustAnchorCell of {root, keyCertSign}. The
3763 Certificate shall be referred to as NewWithNew; then
- 3764 • identify the Certificate in certificationPathCertificates that has the same subjectKeyIdentifier as the NewWithNew
3765 Certificate. The Certificate shall be referred to as NewWithOld. If no such Certificate is found, the Section 13.3.5.9.1 check fails else:
- 3766 • undertake Certification Path Validation on NewWithOld according to the requirements of Section 4.3.2.8. If the Certification Path
3767 Validation fails, the Section 13.3.5.9.1 check fails else:
- 3768 • use the Public Key in NewWithNew to verify the digital signature in NewWithNew. If the digital signature verification fails, the Section
3769 13.3.5.9.1 check fails.

3770 **13.3.5.10 Required Processing on Change of Remote Party Control**

3771 If:

- 3772 • the targetTrustAnchorCell is {supplier, digitalSignature, management}; and
- 3773 • the Entity Identifier in the targetTrustAnchorCell is changed by the replacement; and
- 3774 • the Device is an ESME or a GSME,

3775 then the Device shall:

- 3776 • set the Supplier Name which it displays to the X.500 Distinguished Name in the subject field of the certificate that was used to
3777 populate the targetTrustAnchorCell; and
- 3778 • add an entry in the Billing Data Log with a snapshot cause of 0x00020000 (Change of Supplier) (with the entry added having the same
3779 content as is required on Set Payment Mode Or Tariff change); and
- 3780 • reset the Tariff Block Counter Matrix.

3781 If the targetTrustAnchorCell is {root, keyCertSign, management} and there are any future dated Update Security Credentials or
3782 Activate Firmware Commands held on the Device that have not yet executed, and so the executionDateTime is in the future, then the
3783 Device shall set each executionDateTime to '99991231235959Z'.

3784 If the targetTrustAnchorCell is not {root, keyCertSign, management} and there are any future dated Update Security
3785 Credentials or Activate Firmware Commands held on the Device that:

- 3786 • include replacements for this remotePartyRole; and
 - 3787 • have not yet executed, and so the executionDateTime is in the future;
- 3788 then the Device shall set each executionDateTime to '99991231235959Z'.

3789 If:

- 3790 • the targetTrustAnchorCell is {supplier, digitalSignature, management} or {root, keyCertSign, management};
3791 and
- 3792 • the Entity Identifier in the targetTrustAnchorCell is changed by the replacement

3793 then the Device shall set the execution date-time of any other future dated Commands, that are held on the Device but not yet executed, to
3794 'never', as detailed in Section 9.2. If the deviceType of the Device is gSME then the Device shall also delete any future dated data items that
3795 are pending activation.

3796 If:

- 3797 • remotePartyRole of targetTrustAnchorCell and that of authorisingRemotePartyControl is supplier; and
- 3798 • keyUsage of targetTrustAnchorCell is digitalSignature

3799 then the Device shall:

- 3800 • set all Execution Counters to the value in newRemotePartyFloorSeqNumber;
- 3801 • clear all values from the UTRN Counter Cache; and
- 3802 • place a single value in the UTRN Counter Cache. If newRemotePartySpecialistFloorSeqNumber is present and the seqNumberUsage in that newRemotePartySpecialistFloorSeqNumber is prepaymentTopUp then that value shall be the 32 most significant bits of the seqNumber in the newRemotePartySpecialistFloorSeqNumber. Otherwise the value shall be the 32 most significant bits of the newRemotePartyFloorSeqNumber.

3806 If (remotePartyRole of authorisingRemotePartyControl is not supplier) but (targetTrustAnchorCell is {supplier, digitalSignature, management}) then there should be an instance of otherRemotePartySeqNumberChanges where
3807 remotePartyRole is supplier in the Command. Using the values in that instance of otherRemotePartySeqNumberChanges or the
3808 values zero if there is no such instance, the Device shall:

- 3810 • set all Execution Counters to the value in otherRemotePartyFloorSeqNumber;
- 3811 • clear all values from the UTRN Counter Cache; and
- 3812 • place a single value in the UTRN Counter Cache. If newRemotePartySpecialistFloorSeqNumber is present and the seqNumberUsage in that newRemotePartySpecialistFloorSeqNumber is prepaymentTopUp then that value shall be the 32 most significant bits of the seqNumber in the newRemotePartySpecialistFloorSeqNumber. Otherwise the value shall be the 32 most significant bits of the otherRemotePartyFloorSeqNumber.

3816 13.3.5.11 The `@UpdateSecurityCredentials.CommandPayload`, `@UpdateSecurityCredentials.ResponsePayload` and
3817 `@UpdateSecurityCredentials.AlertPayload` **structure definition**

3818 Each instance of `@UpdateSecurityCredentials.CommandPayload`, `@UpdateSecurityCredentials.ResponsePayload` and of
3819 `@UpdateSecurityCredentials.AlertPayload` shall be an octet string containing the DER encoding of the populated structure defined in
3820 this Section 13.3.4, which specifies the structure in ASN.1.

3821 The structure of Certificate shall be as defined in ASN.1 in IETF RFC 5912. Note that the Certificate structures within IETF RFC 5912
3822 begin after the phrase ‘Certificate- and CRL-specific structures begin here’.

```
3823 UpdateSecurityCredentials DEFINITIONS ::= BEGIN
3824
3825 CommandPayload ::= SEQUENCE
3826 {
3827   -- Provide details to allow the Device to identify the Remote Party Role authorising
3828   -- this Command, check whether the rest of the payload is allowable, prevent replay attacks
3829   -- and allow counters / counter caches on the Device to be reset, if the Command changes the Remote Party
3830   -- in control.
3831   -- The Remote Party authorising the Command is that party which generated the KRP Signature (or the Access Control Broker
3832   -- if there is no KRP Signature)
3833
3834   authorisingRemotePartyControl           AuthorisingRemotePartyControl,
3835
3836   -- One TrustAnchorReplacement structure is required for each Trust Anchor Cell that is to be updated
3837
3838   replacements                         SEQUENCE OF TrustAnchorReplacement,
3839
3840   -- Provide the certificates needed to undertake Certification Path Validation of the new
3841   -- end entity certificate against the root public key held on the Device. The number of these may be less
3842   -- than the number of replacement certificates (e.g. a supplier may replace all of its certificates but
3843   -- may only need to supply one Certification Authority Certificate to link them all back to the root public
3844   -- key as currently stored on the Device.
3845
3846   certificationPathCertificates        SEQUENCE OF Certificate,
3847
3848   -- If the Command is to be future dated, specify the date-time at which the certificate replacement is to happen
3849
3850   executionDateTime                  GeneralizedTime OPTIONAL
3851
3852 }
3853
3854 ResponsePayload ::= SEQUENCE
```

```
3855 {
3856     -- if the Command is future dated, the Response will not have any details of execution (those will be in the subsequent alert)
3857
3858     commandAccepted           NULL,
3859
3860     -- if the Command is for immediate execution, the Response will detail the outcomes
3861
3862     executionOutcome          ExecutionOutcome OPTIONAL
3863
3864 }
3865
3866 AlertPayload ::=           SEQUENCE
3867 {
3868     -- specify the Alert Code
3869     alertCode                INTEGER(0..4294967295),
3870
3871     -- specify the date-time of execution
3872     executionDateTime         GeneralizedTime,
3873
3874
3875     -- detail what happened when the future dated Command was executed
3876
3877     executionOutcome          ExecutionOutcome
3878
3879 }
3880
3881 ExecutionOutcome ::=        SEQUENCE
3882 {
3883     -- Provide details of the corresponding Command that may not be in the standard GBCS message header. Specifically the
3884     -- mode in which the Command was invoked, the Originator Counter in the original Command and the resulting changes to any
3885     -- replay counters held on the Device
3886
3887     authorisingRemotePartySeqNumber    SeqNumber,
3888     credentialsReplacementMode       CredentialsReplacementMode,
3889     remotePartySeqNumberChanges     SEQUENCE OF RemotePartySeqNumberChange,
3890
3891     -- For each replacement in the Command, detail the outcome and impacted parties
3892
3893     replacementOutcomes           SEQUENCE OF ReplacementOutcome
3894
3895 }
3896
3897 AuthorisingRemotePartyControl ::=   SEQUENCE
```

```
3898 {  
3899   -- Specify the replacement mode so that the Device can check that the Remote Party Role is allowed to  
3900   -- authorise this type of replacement and that all replacements in the payload are allowed within this  
3901   -- replacement mode  
3902  
3903   credentialsReplacementMode          CredentialsReplacementMode,  
3904  
3905   -- Only if credentialsReplacementMode = anyByContingency, provide the symmetric key to decrypt  
3906   -- the Contingency Public Key in the (root, digitalSignature, management) Trust Anchor Cell  
3907  
3908   plaintextSymmetricKey           [0] IMPLICIT OCTET STRING OPTIONAL,  
3909  
3910   -- Specify whether the time based checks as part of any Certificate Path Validation should be applied  
3911  
3912   applyTimeBasedCPVChecks        [1] IMPLICIT INTEGER {apply(0), disapply(1)} DEFAULT apply,  
3913  
3914   -- Identify which of the Public Keys on the Device is to be used in checking KRP Signature  
3915   -- 'authorisingRemotePartyTACellIdentifier' can only be omitted when  
3916   -- the access control broker is updating its own credentials. In all other cases it is mandatory.  
3917  
3918   authorisingRemotePartyTACellIdentifier [2] IMPLICIT TrustAnchorCellIdentifier OPTIONAL,  
3919  
3920   -- Specify the Originator Counter for the Remote Party Applying KRP Signature, or (for the  
3921   -- Access Control Broker changing its credentials) the Access Control Broker's Originator Counter.  
3922  
3923   authorisingRemotePartySeqNumber    [3] IMPLICIT SeqNumber,  
3924  
3925   -- If the Command is to effect a change of control, then newTrustAnchorFloorSeqNumber must be included  
3926   -- and will be the value used to prevent replay of Update Security Credentials Commands for the  
3927   -- new controlling Remote Party.  
3928  
3929   newRemotePartyFloorSeqNumber      [4] IMPLICIT SeqNumber OPTIONAL,  
3930  
3931   -- Some Commands on the Device may use a different Originator Counter sequence for Protection Against Replay. At this  
3932   -- version of the GBCS, the only example is the Prepayment Top Up Command on ESME and GSME. The  
3933   -- SpecialistSeqNumber structure allows such Counters to also be reset on change of control.  
3934  
3935   newRemotePartySpecialistFloorSeqNumber [5] IMPLICIT SEQUENCE OF SpecialistSeqNumber OPTIONAL,  
3936  
3937   -- In some cases, one party acting in one Remote Party Role may be replacing certificates for a different Remote Party Role.  
3938   -- In some cases, sequence counters need also to be reset for those other Remote Party Role(s)  
3939  
3940   otherRemotePartySeqNumberChanges  [6] IMPLICIT SEQUENCE OF RemotePartySeqNumberChange OPTIONAL
```

```
3941 }
3942
3943 RemotePartySeqNumberChange ::= SEQUENCE
3944 {
3945   otherRemotePartyRole           RemotePartyRole,
3946   otherRemotePartyFloorSeqNumber SeqNumber,
3947   newRemotePartySpecialistFloorSeqNumber SEQUENCE OF SpecialistSeqNumber OPTIONAL
3948 }
3949
3950 SpecialistSeqNumber ::= SEQUENCE
3951 {
3952   -- Specify the usage of the SeqNumber
3953   seqNumberUsage                SeqNumberUsage,
3954
3955   -- Specify the associated SeqNumber
3956   seqNumber                     SeqNumber
3957 }
3958
3959 SeqNumberUsage ::= INTEGER
3960 {
3961   -- Define the full set of discrete usages on a Device. The only specialist
3962   -- counter is for Prepayment Top Up (which is set independently of other counters). This may only be
3963   -- included when changing Supplier Security Credentials on an ESME or GSME.
3964
3965   prepaymentTopUp               (0)
3966 }
3967
3968 SeqNumber ::= INTEGER (0..9223372036854775807)
3969
3970
3971 TrustAnchorReplacement ::= SEQUENCE
3972 {
3973   -- Provide the new end entity certificate
3974
3975   replacementCertificate        Certificate,
3976
3977   -- Specify where it is to go (specifically which Trust Anchor Cell is to have its details replaced using
3978   -- the new end entity certificate)
3979
3980   targetTrustAnchorCell         TrustAnchorCellIdentifier
3981 }
3982
3983
```

```
3984 ReplacementOutcome ::=          SEQUENCE
3985 {
3986   affectedTrustAnchorCell           TrustAnchorCellIdentifier,
3987   statusCode                         StatusCode,
3988
3989   -- The GBCS Certificate requirements mean that the subjectUniqueID attribute in the subject field of a certificate will always
3990   -- contain the 64 bit unique number that equates to Entity Identifier. existingSubjectUniqueID should be set
3991   -- accordingly based on the contents of the Trust Anchor Cell prior to Command processing.
3992
3993   existingSubjectUniqueID           OCTET STRING,
3994
3995   -- The GBCS Certificate requirements mean that subjectKeyIdentifier attributes will all be 8 byte SHA-1 Hashes.
3996   -- existingSubjectKeyIdentifier should be set accordingly based on the contents of the Trust Anchor Cell prior to
3997   -- Command processing.
3998
3999   existingSubjectKeyIdentifier      OCTET STRING,
4000
4001   -- The subjectUniqueID in the subject field of the certificate in this TrustAnchorReplacement
4002
4003   replacingSubjectUniqueID         OCTET STRING,
4004
4005   -- The subjectKeyIdentifier in the certificate in this TrustAnchorReplacement
4006
4007   replacingSubjectKeyIdentifier    OCTET STRING
4008 }
4009
4010 TrustAnchorCellIdentifier ::=        SEQUENCE
4011 {
4012   -- Which Remote Party Role does this Cell relate to?
4013
4014   trustAnchorCellRemotePartyRole    RemotePartyRole,
4015
4016   -- To what cryptographic use can the Public Key in this Cell be put? Some Remote Party Roles
4017   -- (e.g. supplier) can have more than one Public Key on a Device and each one would only have
4018   -- a single cryptographic use.
4019
4020   trustAnchorCellKeyUsage          KeyUsage,
4021
4022   -- trustAnchorCellUsage is to allow for multiple Public Keys of the same keyUsage for the same Remote
4023   -- Party Role. It will be absent except where used to refer to the Supplier Key
4024   -- Agreement Key used solely in relation to validating Supplier generated MACs on Prepayment Top Up
4025   -- transactions
4026
```

```
4027 trustAnchorCellUsage           CellUsage DEFAULT management
4028 }
4029
4030 CellUsage ::=                  INTEGER {management(0), prePaymentTopUp(1)}
4031
4032 RemotePartyRole ::=           INTEGER
4033 {
4034   -- Define the full set of Remote Party Roles in relation to which a Device may need to undertake
4035   -- processing. Note that most Devices will only support a subset of these.
4036
4037   root                      (0),
4038   recovery                   (1),
4039   supplier                  (2),
4040   networkOperator            (3),
4041   accessControlBroker       (4),
4042   transitionalCoS           (5),
4043   wanProvider                (6),
4044   issuingAuthority          (7),    -- Devices will receive such Certificates but they do not need to store
4045                           -- them over an extended period
4046
4047   -- The 'other' RemotePartyRole is for a party whose role does not allow it to invoke any Device function apart from
4048   -- UpdateSecurityCredentials. This is to allow for Device functionality to be locked out of usage until a valid
4049   -- Remote Party can be identified e.g. where roles cannot be fixed until a Device is brought in to operation
4050
4051   other                      (127)
4052
4053 }
4054
4055   -- KeyUsage is only repeated here for clarity. It is defined in RFC 5912
4056
4057 KeyUsage ::=                  BIT STRING
4058 {
4059   -- Define valid uses of Public Keys held by Devices in their Trust Anchor Cells.
4060
4061   digitalSignature           (0),
4062   contentCommitment          (1),    -- not valid for GBCS compliant transactions
4063   keyEncipherment            (2),    -- not valid for GBCS compliant transactions
4064   dataEncipherment           (3),    -- not valid for GBCS compliant transactions
4065   keyAgreement               (4),
4066   keyCertSign                (5),
4067   cRLSign                    (6),
4068   encipherOnly                (7),    -- not valid for GBCS compliant transactions
4069   decipherOnly                (8)     -- not valid for GBCS compliant transactions
```

```
4070 }
4071
4072 CredentialsReplacementMode ::= INTEGER
4073 {
4074 -- Define the valid combinations as to which Remote Party Roles can replace which kinds of Trust Anchors.
4075
4076 -- Normal operational replacement modes
4077 rootBySupplier (0),
4078 rootByWanProvider (1),
4079 supplierBySupplier (2),
4080 networkOperatorByNetworkOperator (3),
4081 accessControlBrokerByACB (4),
4082 wanProviderByWanProvider (5),
4083 transCoSByTransCos (6),
4084 supplierByTransCos (7),
4085
4086 -- Recovery modes
4087 anyExceptAbnormalRootByRecovery (8),
4088 anyByContingency (9)
4089 }
4090
4091 -- The GBCS only allows for a constrained set of Trust Anchor Cell operations and so the list of possible outcomes
4092 -- is more limited than in RFC 5934. The list below is that more constrained subset
4093
4094 StatusCode ::= ENUMERATED {
4095
4096 success (0),
4097
4098 -- badCertificate is used to indicate that the syntax for one or more certificates is invalid.
4099
4100 badCertificate (5),
4101
4102 -- noTrustAnchor is used to indicate that the authorityKeyIdentifier does not identify the public key of a
4103 -- trust anchor or a certification path that terminates with an installed trust anchor
4104
4105 noTrustAnchor (10),
4106
4107 -- insufficientMemory indicates that the update could not be processed because the Device did not
4108 -- have sufficient memory
4109
4110 insufficientMemory (17),
4111
4112 -- contingencyPublicKeyDecrypt indicates that the update could not be processed because an error occurred while
```

```

4113 -- decrypting the contingency public key.
4114
4115 contingencyPublicKeyDecrypt          (22),
4116
4117 -- trustAnchorNotFound indicates that a change to a trust anchor was requested, but the referenced trust anchor
4118 -- is not represented in the Trust Anchor Cell.
4119
4120 trustAnchorNotFound          (25),
4121
4122 -- resourcesBusy indicates that the resources necessary to process the replacement are not available at the
4123 -- present time, but the resources might be available at some point in the future.
4124
4125 resourcesBusy          (30),
4126
4127 -- other indicates that the update could not be processed, but the reason is not covered by any of the assigned
4128 -- status codes. Use of this status code SHOULD be avoided.
4129
4130 other          (127) }
4131
4132 END

```

4133 13.3.5.12 Requirements for AuthorisingRemotePartyControl elements - informative

4134 All bar two parts of the AuthorisingRemotePartyControl structure are optional. This section summarises when each of the optional
 4135 elements needs to be present.

AuthorisingRemotePartyControl element	Notes
credentialsReplacementMode	Always required
plaintextSymmetricKey	Only required if credentialsReplacementMode = anyByContingency (when it is always required)
applyTimeBasedCPVChecks	Only required if the Device is to ignore time when undertaking Certification Path Validation, in which case it needs to have the value 'disapply'
authorisingRemotePartyTACellIdentifier	For a Communications Hub, always present. For all other Devices, always present unless the Access Control Broker is replacing its own credentials (in which case it should be omitted)
authorisingRemotePartySeqNumber	Always required

AuthorisingRemotePartyControl element	Notes
newRemotePartyFloorSeqNumber	If the Command is to effect a change of control, then newTrustAnchorFloorSeqNumber should be included. It can be present in all other situations
newRemotePartySpecialistFloorSeqNumber	Only required on Change of Supplier where the new Supplier has decided to use a different sequence of Originator Counters for prepayment top ups.
otherRemotePartySeqNumberChanges	Should be present if one role (e.g. recovery, transitionalCoS) is changing credentials for another role or roles (e.g. supplier). In such cases, this should be present to set Protection Against Replay counters for that other role or roles

4136

Table 13.3.5.12: Requirements for AuthorisingRemotePartyControl elements

13.4 CS02c Issue Security Credentials

13.4.1 Description

This section covers the creation, validation and processing of (i) Issue Security Credentials Commands and (ii) Responses to such Commands.

13.4.2 Use Case Cross References

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Command and Response
Message Type Category	SME.C.C
Capable of future dated invocation?	No
Protection Against Replay Required?	Yes
SEC User Gateway Services Schedule (Service Request) Reference	6.17
Valid Target Device(s)	ESME / GSME / GPF / CHF
Valid Business Originator role(s) for Command invocation (and so, for DLMS COSEM Commands, which Application Association is to be used for delivery of the Command to the Device) [Remote Party Messages Only]	Supplier (for Devices other than CHF) WAN Provider (for CHF Devices only)
Valid Response Recipient role(s) (only for Messages Authorised by the Access Control Broker on behalf of parties not known to the Device) [Remote Party Messages Only]	None
Valid initiating Device type(s) [HAN Only Messages]	None
Protocol	ASN.1

Table 13.4.2: Use Case Cross References for Issue Security Credential Details Command and Response

13.4.3 Construction of Commands

Issue Security Credentials Command Payloads shall be constructed as specified in Section 13.4.7 and Cryptographic Protection I and Cryptographic Protection II shall be applied as required for a Command of Message Category SME.C.C.

13.4.4 Device processing of Commands and Response handling

The Device receiving an Issue Security Credentials Command shall undertake processing steps in the sequence defined in this Section 13.4.4.

In processing an Issue Security Credentials Command, the Device shall:

72. undertake Command Authenticity and Integrity Verification as required for a Command of Message Category SME.C.C. The Security Credentials used to verify Cryptographic Protection I shall be:

- o those held in the {wANProvider, digitalSignature, management} Trust Anchor Cell, if the target Device's deviceType equals communicationsHubCommunicationsHubFunction;

- 4156 ○ those held in the {supplier, digitalSignature, management} Trust Anchor
 4157 Cell, if the target Device's deviceType does not equal
 4158 communicationsHubCommunicationsHubFunction;
- 4159 73. validate that the value of keyUsage in CommandPayload is either digitalSignature
 4160 only or keyAgreement only. If this validation fails then the Device shall set
 4161 issueCredentialsResponseCode to invalidKeyUsage, and process from step 6;
- 4162 74. generate a Private-Public Key Pair and store the Private Key so generated in the
 4163 Pending Private Key Cell determined by the value of keyUsage in CommandPayload.
 4164 If the step fails then the Device shall set issueCredentialsResponseCode to
 4165 keyPairGenerationFailed, and process from step 6;
- 4166 75. with the ASN.1 terms in this step (that are not defined in this Section 13.4.4) having the
 4167 meaning of IETF RFC 2986³⁰; generate a CertificationRequest which:
- 4168 ○ complies with the requirements of IETF RFC2986 and IETF RFC 5912;
 - 4169 ○ is DER encoded, in line with the recommendation of IETF RFC 5967³¹;
 - 4170 ○ has subjectPublicKey set to the bit string representation of the Public Key
 4171 generated in step 3;
 - 4172 ○ incorporates an extensionRequest identified by id-ce-keyUsage which shall
 4173 contain the keyUsage value specified in CommandPayload;
 - 4174 ○ incorporates an extensionRequest identified by id-ce-subjectAltName
 4175 which shall contain a single GeneralName of type OtherName that is further sub-
 4176 typed as a HardwareModuleName (id-on-HardwareModuleName) as defined in
 4177 IETF RFC 4108. The hwSerialNum field shall be set to the Device's Entity
 4178 Identifier; and
 - 4179 ○ has the signature generated using the Private Key generated in step 3;
- 4180 76. if the generation of CertificationRequest is not successful then the Device shall
 4181 set issueCredentialsResponseCode to cRProductionFailed;
- 4182 77. create a Response according to the requirements of Section 13.4.7, apply the Response
 4183 Cryptographic Protection required for a Response of Message Category SME.C.C, and
 4184 send the Response.

4185 13.4.5 Response Processing

4186 Response Recipient Verification may be undertaken as specified in this GBCS for a
 4187 Response of Message Category SME.C.C. The issueCredentialsResponseCode field,
 4188 where present in the Response, may be decoded according to the ASN.1 definitions at
 4189 Section 13.4.6.

³⁰ <https://tools.ietf.org/html/rfc2986>

³¹ <https://tools.ietf.org/html/rfc5967>

4190 13.4.6 Issue Security Credentials Command and Response payloads – structure definition

4191 Each instance of `@IssueSecurityCredentials.CommandPayload` and of `@IssueSecurityCredentials.ResponsePayload` shall be
 4192 an octet string containing the DER encoding of the populated structure defined in this Section 13.4.6 which specifies the structure in ASN.1
 4193 notation.

```

4194 IssueSecurityCredentials DEFINITIONS ::= BEGIN
4195
4196   CommandPayload ::=                               SEQUENCE
4197   {
4198     -- specify the keyUsage to which the generated key-pair will be put, if subsequently authorised
4199     keyUsage                                KeyUsage
4200   }
4201
4202   ResponsePayload ::=                           CHOICE
4203   {
4204     -- if the Command was successful, provide the generated Certification Request. CertificationRequest shall
4205     -- be as defined in ASN.1 by IETF RFC 5912. For reference, it is in the section headed 'ASN.1 Module for RFC 2986',
4206     certificationRequest                      CertificationRequest,
4207
4208     -- if the Command was unsuccessful, detail the failure
4209
4210     issueCredentialsResponseCode           IssueCredentialsResponseCode
4211   }
4212
4213
4214   -- KeyUsage is only repeated here for ease of reference. It is defined in IETF RFC 5912
4215
4216   KeyUsage ::=                               BIT STRING
4217   {
4218     -- Define valid uses of Public Keys held by Devices in their Trust Anchor Cells.
4219
4220     digitalSignature                         (0),
4221     contentCommitment                        (1),    -- not valid for GBCS compliant transactions
4222     keyEncipherment                          (2),    -- not valid for GBCS compliant transactions
4223     dataEncipherment                         (3),    -- not valid for GBCS compliant transactions
4224     keyAgreement                            (4),
4225     keyCertSign                             (5),    -- not valid for this Use Case
4226     cRLSign                                 (6),    -- not valid for this Use Case
4227     encipherOnly                            (7),    -- not valid for GBCS compliant transactions
4228     decipherOnly                            (8)     -- not valid for GBCS compliant transactions
4229
  
```

```

4230 }
4231
4232 IssueCredentialsResponseCode ::= INTEGER
4233 {
4234     invalidKeyUsage          (1),
4235     keyPairGenerationFailed (2),
4236     cRProductionFailed      (3)
4237 }
4238
4239
4240 END

```

4241 13.4.7 Constructing the @IssueSecurityCredentials.CommandPayload and of 4242 @IssueSecurityCredentials.ResponsePayload

4243 @IssueSecurityCredentials.CommandPayload shall have the structure defined in Section 13.4.6, and the Remote Party constructing
4244 the Command shall populate with values according to Table 13.4.7a.

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
@IssueSecurityCredentials.CommandPayload	SEQUENCE			
keyUsage	BIT STRING	Either digitalSignature (0) only, Or keyAgreement (4) only	Mandatory	Only one or the other is valid

4245 Table 13.4.7a: @IssueSecurityCredentials.CommandPayload population

4246 @IssueSecurityCredentials.ResponsePayload shall have the structure defined in Section 13.4.6, and the Device constructing the
4247 Response shall populate with values according to Table 13.4.7b.

4248

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
@IssueSecurityCredentials.ResponsePayload	CHOICE			
certificationRequest	See IETF RFC 5912	The Certification Request produced according to the requirements of Section 13.4.4.	Mandatory	Mandatory if certificationRequest successfully produced
issueCredentialsResponseCode	INTEGER	Shall be populated according to the processing defined in Section 13.4.4	Mandatory	Mandatory if certificationRequest is not successfully produced

4249 Table 13.4.7b: @IssueSecurityCredentials.ResponsePayload population

4250 13.5 CS02d Update Device Certificates on Device

4251 13.5.1 Description

4252 This Section 13.5 covers the creation, validation and processing of (i) Update Device
 4253 Certificates on Device, Commands and (ii) Responses to such Commands.

4254 13.5.2 Use Case Cross References

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Command and Response
Message Type Category	SME.C.C
Capable of future dated invocation?	No
Protection Against Replay Required?	Yes
SEC User Gateway Services Schedule (Service Request) Reference	6.15
Valid Target Device(s)	ESME / GSME / GPF / CHF
Valid Business Originator role(s) for Command invocation (and so, for DLMS COSEM Commands, which Application Association is to be used for delivery of the Command to the Device) [Remote Party Messages Only]	Supplier (for Devices other than CHF) WAN Provider (for CHF Devices only)
Valid Response Recipient role(s) (only for Messages Authorised by the Access Control Broker on behalf of parties not known to the Device) [Remote Party Messages Only]	None
Valid initiating Device type(s) [HAN Only Messages]	None
Protocol	ASN.1

Table 13.5.2: Use Case Cross References for Update Device Certificate on Device, Command and Response

4255 13.5.3 Construction of Commands

4256 Update Device Certificate on Device Command Payloads shall be constructed as specified
 4257 in Section 13.5.7, and Cryptographic Protection I and Cryptographic Protection II shall be
 4258 applied as required for a Command of Message Category SME.C.C.

4259 13.5.4 Device processing of Commands and Response handling

4260 The Device receiving an Update Device Certificate on Device Command shall undertake
 4261 processing steps in the sequence defined in this Section 13.5.4.

4262 In processing an Update Device Certificate on Device Command, the Device shall:

4263 78. **undertake Command Authenticity and Integrity Verification as required for a Command of**
 4264 **Message Category SME.C.C. The Security Credentials used to verify Cryptographic**
 4265 **Protection I shall be:**

- 4266 **o those held in the {wANProvider, digitalSignature, management} Trust**
 4267 **Anchor Cell, if the target Device's deviceType equals**
 4268 **communicationsHubCommunicationsHubFunction; or**

- 4269 ○ those held in the {supplier, digitalSignature, management} Trust Anchor
 4270 Cell, if the target Device's deviceType does not equal
 4271 communicationsHubCommunicationsHubFunction.
- 4272 79. establish the values of keyUsage, subjectPublicKey and hwSerialNum in
 4273 certificate in the CommandPayload. If any of the values cannot be established then the
 4274 Device shall set updateDeviceCertResponseCode to invalidCertificate, and
 4275 process from step 10;
- 4276 80. validate that hwSerialNum established at step 2 is the Device's Entity Identifier. If this
 4277 validation fails then the Device shall set updateDeviceCertResponseCode to
 4278 wrongDeviceIdentity, and process from step 10;
- 4279 81. validate that keyUsage established at step 2 is either digitalSignature only or
 4280 keyAgreement only. If this validation fails then the Device shall set
 4281 updateDeviceCertResponseCode to invalidKeyUsage, and process from step 10;
- 4282 82. validate that the Device holds a Pending Private Key for the keyUsage as established at
 4283 step 2. If this validation fails then the Device shall set
 4284 updateDeviceCertResponseCode to noCorrespondingKeyPairGenerated, and
 4285 process from step 10;
- 4286 83. validate that subjectPublicKey established at step 2 is the bit string representation of
 4287 the Public Key corresponding to the Pending Private Key identified at step 5. If this
 4288 validation fails then the Device shall set updateDeviceCertResponseCode to
 4289 wrongPublicKey, and process from step 10;
- 4290 84. store certificate. If this step fails then the Device shall set
 4291 updateDeviceCertResponseCode to certificateStorageFailed, and process
 4292 from step 10;
- 4293 85. set the Current Private Key to have the value of the Pending Private Key for the
 4294 keyUsage established at step 2. If this step fails then the Device shall set
 4295 updateDeviceCertResponseCode to privateKeyChangeFailed, and process
 4296 from step 10;
- 4297 86. set updateDeviceCertResponseCode to success; and
- 4298 87. create a Response according to the requirements of Section 13.5.7, apply the Response
 4299 Cryptographic Protection required for a Response of Message Category SME.C.C, and
 4300 send the Response.
- 4301 If all steps were successful and this was a change of digitalSignature certificate,
 4302 the Response shall be signed using the private key corresponding to the new certificate.
 4303 If there was a failure, the Response shall be signed using the private key corresponding to
 4304 the pre-existing key pair.
- 4305 Once the Pending Private Key becomes the Current Private Key, the Device will be using
 4306 the new Private Key and this will affect all Remote Parties interacting with the Device;
 4307 specifically they will need to use the new Certificate corresponding to the Private Key now in
 4308 use.

4309 13.5.5 Response Processing

4310 Response Recipient Verification may be undertaken as specified in this GBCS for a
 4311 Response of the relevant Message Category. The updateDeviceCertResponseCode
 4312 field may be decoded according to the ASN.1 definitions at Section 13.5.6.

4313 If this was a change of digitalSignature certificate, the public key to be used to verify
4314 the Device's signature is dependent on the value of updateDeviceCertResponseCode.
4315 If updateDeviceCertResponseCode is success then the Private Key used for Signing
4316 will have changed. If updateDeviceCertResponseCode is other than success, the
4317 Private Key used for Signing will not have changed.

13.5.6 Update Device Certificate on Device Command and Response payloads – structure definition

Each instance of `@UpdateDeviceCertificateonDevice.CommandPayload` and of `@UpdateDeviceCertificateonDevice.ResponsePayload` shall be an octet string containing the DER encoding of the populated structure defined in this Section 13.5.6, which specifies the structure in ASN.1 notation.

```

4324 UpdateDeviceCertificateonDevice DEFINITIONS ::= BEGIN
4325
4326 CommandPayload ::= Certificate
4327   -- provide the certificate which the Device is to store
4328   -- the ASN.1 specification of certificate shall be as defined in IETF RFC 5912
4329 for IETF RFC 5280
4330
4331 ResponsePayload ::= UpdateDeviceCertResponseCode
4332
4333   -- if the Command was unsuccessful, detail the failure; otherwise respond with
4334 success
4335
4336 UpdateDeviceCertResponseCode ::= INTEGER
4337 {
4338   success                               (0),
4339   invalidCertificate                     (1),
4340   wrongDeviceIdentity                   (2),
4341   invalidKeyUsage                      (3),
4342   noCorrespondingKeyValuePairGenerated (4),
4343   wrongPublicKey                       (5),
4344   certificateStorageFailed             (6),
4345   privateKeyChangeFailed               (7)
4346 }
4347
4348 END

```

13.5.7 Constructing the

`@UpdateDeviceCertificateonDevice.CommandPayload` and
`@UpdateDeviceCertificateonDevice.ResponsePayload`

4352 @UpdateDeviceCertificateonDevice.CommandPayload shall have the structure
4353 defined in Section 13.5.6, and the Remote Party constructing the Command shall populate
4354 with values according to Table 13.5.7a.

4355

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value
@UpdateDeviceCertificateonDevice.CommandPayload			
Certificate	See IETF RFC 5912	A new Device Certificate that the Device is to process	Mandatory

Table 13.5.7a: @UpdateDeviceCertificateonDevice.CommandPayload population

4356 @UpdateDeviceCertificateonDevice.ResponsePayload shall have the structure
 4357 defined in Section 13.5.6, and the Device constructing the Response shall populate with
 4358 values according to Table 13.5.7b.

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value
@UpdateDeviceCertificateonDevice.ResponsePayload			
UpdateDeviceCertResponseCode	INTEGER	Shall be populated according to the processing defined in Section 13.5.4	Mandatory

13.6 Table 13.5.7b:

@UpdateDeviceCertificateonDevice.ResponsePayload

populationCS02e Provide Device Certificates from Device

13.6.1 Description

This section covers the creation, validation and processing of (i) Provide Device Certificates from Device Commands and (ii) Responses to such Commands.

13.6.2 Use Case Cross References

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Command & Response
Message Type Category	Variant Message
Capable of future dated invocation?	No
Protection Against Replay Required?	No
SEC User Gateway Services Schedule (Service Request) Reference	6.24
Valid Target Device(s)	ESME / GSME / GPF / CHF
Valid Business Originator role(s) for Command invocation (and so, for DLMS COSEM Commands, which Application Association is to be used for delivery of the Command to the	Supplier (for Devices other than CHF) WAN Provider (for CHF Devices only)

Device) [Remote Party Messages Only]	
Valid Response Recipient role(s) (only for Messages Authorised by the Access Control Broker on behalf of parties not known to the Device) [Remote Party Messages Only]	None
Valid initiating Device type(s) [HAN Only Messages]	None
Protocol	ASN.1

4367 Table 13.6.2: Use Case Cross References for Provide Device Certificates from Device Command
 4368 and Response

13.6.3 Construction of Commands

4370 Provide Device Certificates from Device Command Payloads shall be constructed as
 4371 specified in Section 13.6.7 and Cryptographic Protection I and Cryptographic Protection II
 4372 shall be applied as required for a Command of Message Category SME.C.C.

13.6.4 Device processing of Commands and Response handling

4374 The Device receiving a Provide Device Certificates from Device Command shall undertake
 4375 processing steps in the sequence defined in this Section 13.6.4.

4376 In processing a Provide Device Certificates from Device Command, the Device shall:

- 4377 88. **undertake Command Authenticity and Integrity Verification as required for a Command of**
4378 Message Category SME.C.C, except that Cryptographic Protection II shall not be
4379 verified. The Security Credentials used to verify Cryptographic Protection I shall be:
 - 4380 ○ **those held in the {wANProvider, digitalSignature, management} Trust**
4381 Anchor Cell, if the target Device's deviceType equals
4382 communicationsHubCommunicationsHubFunction; or
 - 4383 ○ **those held in the {supplier, digitalSignature, management} Trust Anchor**
4384 Cell, if the target Device's deviceType does not equal
4385 communicationsHubCommunicationsHubFunction;
- 4386 89. **validate that keyUsage in CommandPayload is either digitalSignature only or**
4387 keyAgreement only. If this validation fails then the Device shall set
4388 provideDeviceCertResponseCode to invalidKeyUsage, and process from step 5;
- 4389 90. **confirm that the Device holds a certificate which (1) is for the keyUsage identified at**
4390 step 2, (2) contains in hwSerialNum a value equal to the Device's Entity Identifier and
4391 (3) contains in subjectPublicKey the bit string representation of the Public Key
4392 corresponding to the Current Private Key for this keyUsage. If this validation fails then
4393 the Device shall set provideDeviceCertResponseCode to noCertificateHeld,
4394 and process from step 5;
- 4395 91. **retrieve the certificate identified in step 3. If this step fails then the Device shall set**
4396 provideDeviceCertResponseCode to certificateRetrievalFailure, and
4397 process from step 5;
- 4398 92. **create a Response according to the requirements of Section 13.6.7, apply the Response**
4399 Cryptographic Protection required for a Response of Message Category SME.C.C, and
4400 send the Response.

4401 **13.6.5 Response Processing**

4402 Response Recipient Verification may be undertaken as specified in this GBCS for a
4403 Response of the Message Category SME.C.C. The provideDeviceCertResponseCode
4404 field may be decoded according to the ASN.1 definitions at Section 13.6.6.

4405 13.6.6 Provide Device Certificates from Device Command and Response payloads – structure definition

4406 Each instance of `@ProvideDeviceCertificateFromDevice.CommandPayload` and of
4407 `@ProvideDeviceCertificateFromDevice.ResponsePayload` shall be an octet string containing the DER encoding of the populated
4408 structure defined in this Section 13.6.6 which specifies the structure in ASN.1 notation.

```
4409 ProvideDeviceCertificateFromDevice DEFINITIONS ::= BEGIN
4410
4411     CommandPayload ::=          SEQUENCE
4412     {
4413         -- specify the KeyUsage of the Certificate to be returned
4414         keyUsage           KeyUsage
4415
4416     }
4417
4418     ResponsePayload ::=          CHOICE
4419
4420     {
4421         -- if the Command was successful, provide the certificate
4422         certificate        Certificate,
4423
4424         -- if the Command was unsuccessful, detail the failure
4425
4426         provideDeviceCertResponseCode   ProvideDeviceCertResponseCode
4427     }
4428
4429
4430 -- KeyUsage is only repeated here for ease of reference. It is defined in RFC 5912
4431
4432     KeyUsage ::=          BIT STRING
4433     {
4434         -- Define valid uses of Public Keys held by Devices in their Trust Anchor Cells.
4435
4436         digitalSignature          (0),
4437         contentCommitment         (1),    -- not valid for GBCS compliant transactions
4438         keyEncipherment          (2),    -- not valid for GBCS compliant transactions
4439         dataEncipherment         (3),    -- not valid for GBCS compliant transactions
4440         keyAgreement             (4),
4441         keyCertSign              (5),    -- not valid for this Use Case
4442         cRLSign                  (6),    -- not valid for this Use Case
4443         encipherOnly             (7),    -- not valid for GBCS compliant transactions
4444         decipherOnly             (8)     -- not valid for GBCS compliant transactions
```

```

4445 }
4446
4447 ProvideDeviceCertResponseCode ::= INTEGER
4448 {
4449     invalidKeyUsage          (1),
4450     noCertificateHeld        (2),
4451     certificateRetrievalFailure (3)
4452 }
4453
4454
4455 END

```

4456 13.6.7 Constructing the @ProvideDeviceCertificateFromDevice.CommandPayload and 4457 @ProvideDeviceCertificateFromDevice.ResponsePayload

4458 @ProvideDeviceCertificateFromDevice.CommandPayload shall have the structure defined in Section 13.6.6 and the Remote Party
4459 constructing the Command shall populate with values according to Table 13.6.7a.

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
@ProvideDeviceCertificateFromDevice.CommandPayload	SEQUENCE			
keyUsage	BIT STRING	Either digitalSignature (0) only, Or keyAgreement (4) only	Mandatory	Only one or the other is valid

4460 Table 13.6.7a: @ProvideDeviceCertificateFromDevice.CommandPayload population @ProvideDeviceCertificateFromDevice.Respo
4461 nsePayload shall have the structure defined in Section 13.6.6, and the Device constructing the Response shall populate with values
4462 according to Table 13.6.7b.

4463

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
@ProvideDeviceCertificateFromDevice.ResponsePayload	CHOICE			
Certificate	See IETF RFC 5912	The Device Certificate provided pursuant to Section 13.6.4	Mandatory	Mandatory if certificate successfully produced
provideDeviceCertResponseCode	INTEGER	Shall be populated according to the processing defined in Section 13.5.4	Mandatory	Mandatory if certificate is not successfully produced

4464

Table 13.6.7b: @ProvideDeviceCertificateFromDevice.ResponsePayload population

4465 13.7 Pair-wise Authorisation of Devices

4466 13.7.1 Introduction to pair-wise Authorisation of Devices - 4467 informative

4468 13.7.1.1 *The role of pair-wise Authorisation - informative*

4469 This Section 13.7 includes the Use Cases related to the Authorisation (and the removal of
 4470 Authorisation) for pair-wise, secure application layer interaction between two Devices on the
 4471 same SMHAN. It also covers the related Use Cases for backing up and restoring the GPF
 4472 Device Log.

4473 The process of authorising two Devices to communicate is referred to as 'Joining'³².
 4474 Removal of such authorisation is referred to as 'Un-joining'. Correspondingly, Remote Party
 4475 Commands are specified in this Section 13.7 for instructing Devices that they are to 'Join' or
 4476 'Unjoin'.

4477 In line with the SMETS and CHTS Device Log requirements, two Devices on a HAN must
 4478 only be capable of interacting at the application layer if they are currently Joined. They must
 4479 not be capable of interacting if (1) they have never been Joined or (2) they have been
 4480 Unjoined.

4481 The application layer interactions between Devices on the same SMHAN must conform to
 4482 the Device Based Access Controls (DBAC) as defined in Section 13.7.3. For example, an
 4483 ESME must not be capable of processing an 'Enable Supply' Command from an IHD or an
 4484 HCALCS.

4485 It is a precondition of Joining that both Devices have been 'White-listed' on to the HAN (as
 4486 per Use Case 'CCS01 Add Device to CHF Device log') so that they are able to communicate
 4487 over the HAN at the network layer (and so have network access). The GPF may be
 4488 configured to be in the CHF's Device Log at manufacture. A Device on a white-list can be
 4489 removed from the white-list. It must then be unable to communicate over the HAN, and so
 4490 unable to interact at the application layer with any Devices to which it was 'Joined'.

4491 In SMETS terminology:

- 4492 • the CHF's Device Log holds the list of currently white-listed Devices on the SMHAN; and
- 4493 • the Device Log on an ESME, GSME, GPF or Type 1 Device holds the Entity Identifiers,
 4494 Device Types and related Security Credentials of other Devices on the HAN to which
 4495 the Device is currently Joined (and so Authorised to interact with at an application layer).

4496 The process of white-listing a Device and its subsequently obtaining network access
 4497 establishes a shared secret key between the Device and the Communications Hub. The
 4498 Gas Proxy Function, which is part of the Communications Hub, uses this shared secret key,
 4499 combined with a Device being entered in to its Device Log, for application layer authorisation.

4500 IHDs and other Type 2 Devices are not required to have a Device Log (as defined in
 4501 SMETS). They are required to store security and related details of the Devices to which they
 4502 are Joined as required by ZSE however (otherwise they would be cryptographically unable
 4503 to understand the information being sent to them by the Joined Devices).

4504 IHDs and other Type 2 Devices can only read application layer information from Devices to
 4505 which they are Joined (either by requesting the information from the Device or by receiving
 4506 information published by the Device). When a PPMID is joined to a GPF the PPMID can
 4507 only read information from the GPF to which it is Joined.

³² This is unrelated to the ZSE meaning of 'joining'

4508 When other types of Device are Joined (e.g. HCALCS, PPMID), they can also exchange
 4509 Commands and Responses at the application layer. For example, an ESME that is Joined
 4510 to an HCALCS can send a Command to the HCALCS to turn its switch on and the HCALCS
 4511 can send a Response saying whether it has done that. A PPMID can send an 'enable
 4512 supply' Command to an ESME etc.

4513 **13.7.1.2 The joining sequence – informative**

4514 There are three types of Join:

- 4515 • **Join Method B**: this is a Join involving a Type 2 Device or a GPF;
- 4516 • **Join Method C**: this is a Join between a GSME and a PPMID; and
- 4517 • **Join Method A**: this is any Join which is not covered by Method B or C.

4518 Except for Method C, all Joins use the ZSE cryptography which requires exchange of
 4519 messages between the two Devices to establish the shared secret that the two Devices will
 4520 need to use. Method C uses the cryptography of Section 4 of this GBCS.

4521 Only certain combinations of Devices can be validly 'Joined'. Table 13.7.1.2 summarises
 4522 valid combinations:

Device Name		ESME	GSME	Comms Hub (CHF)	Comms Hub (GPF)	HCALCS	PPMID	Type 2 (IHD or CAD)
	deviceType	0	1	2	3	4	5	6
ESME	0	Not permitted						
GSME	1	Not permitted	Not permitted					
Comms Hub (CHF)	2	Not permitted	Not permitted	Not permitted				
Comms Hub (GPF)	3	Not permitted	Method B	Not permitted	Not permitted			
HCALCS	4	Method A	Not permitted	Not permitted	Not permitted	Not permitted		
PPMID	5	Method A	Method C	Not permitted	Method B	Not permitted	Not permitted	
Type 2 (IHD or CAD)	6	Method B	Not permitted	Not permitted	Method B	Not permitted	Not permitted	Not permitted

4523 Table 13.7.1.2: Permitted Joins

4524 A Method A Join always involves an ESME and therefore any HAN exchanges required by
 4525 a Method A Join shall always be instigated by the ESME involved. In this context the ESME
 4526 is referred to as the **methodAInitiator**, since it initiates Method A Joins.

4527 The additional step with a Method A Join is that the other Device must first be sent a Join
 4528 Command detailing the ESME with which it is allowed to Join. On receipt, the Device should
 4529 add the ESME details to its Device Log and send a Response accordingly. If, subsequently,
 4530 the Device is asked to undertake key establishment, it must check that the requesting
 4531 Device is in its Device Log.

4532 Only one Device in a Method B Join is remotely instructed. Thus, the HAN exchanges
 4533 required by a Method B Join shall always be instigated by the Device receiving such a
 4534 Command. From Table 13.7.1.2, this is always a GSME or ESME except where a GPF is to

4535 Join to a PPMID, IHD or CAD. Thus, the sequence of a Method B Join is that the ESME /
 4536 GSME / GPF:

- 4537 • is sent a Join Command containing the Entity Identifier of the Device to which it is to
 4538 Join and that other Device's DeviceType;
- 4539 • verifies the cryptographic protection on the Command and checks to make sure it is well
 4540 formed and valid;
- 4541 • updates its Device Log to include details of the new Device;
- 4542 • for an ESME³³, undertakes the key establishment process with the specified Device, as
 4543 per the ZSE 1.2 specification. The constraint that Key Establishment has to involve the
 4544 ZSE Trust Centre shall not be applied by Devices; and
- 4545 • creates and sends a Response detailing the success or otherwise of its actions.

4546 A Method C join does not require exchange of Messages between the two Devices for the
 4547 establishment of the shared secret. Thus the sequence of a Method C Join is that each of
 4548 the GSME and PPMID:

- 4549 • is sent a Join Command containing the Entity Identifier of the Device to which it is to
 4550 Join, that other Device's DeviceType and Key Agreement Certificate;
- 4551 • verifies the cryptographic protection on the Command and does checks to make sure it
 4552 is well formed and valid;
- 4553 • updates its Device Log to include details of the new Device;
- 4554 • checks there is a well-formed Device Certificate in the Command;
- 4555 • optionally calculates the shared secret using the Device Certificate of the other Device
 4556 (which is provided in the Command); and
- 4557 • creates and sends a Response detailing the success or otherwise of its actions.

4558 13.7.1.3 The format of Message Payloads - informative

4559 In common with other GBCS Remote Messages related to the management of Security
 4560 Credentials, the payloads of Commands and Responses defined in this Section 13.7.1.3 are
 4561 specified using ASN.1, with DER encoding to be applied to Command and Response
 4562 payloads.

4563 13.7.2 Device Requirements

4564 All Devices shall:

- 4565 • support the ZSE Key Establishment Cluster as specified in Annex C of the ZSE cluster;
- 4566 • support 'Crypto Suite 2' as defined in the ZSE specification; and
- 4567 • use 'Crypto Suite 2' when undertaking any associated Key Establishment process.

4568 Devices shall not apply any restrictions on the types of Devices used in any associated Key
 4569 Establishment process, except for those specified in the GBCS. Specifically, the ZSE
 4570 constraint requiring Trust Centre involvement shall not be applied (where 'Trust Centre' has
 4571 the meaning defined in ZSE).

4572 An ESME shall be configured to be a ZSE 'Router', as defined in ZSE so that
 4573 communications between the ESME and Devices Joined to the ESME are not reliant on
 4574 availability of the Communications Hub.

³³ The shared secret between the Communications Hub and the Type 2 Device / GSME established when the Device joined the HAN shall be used by the GPF to authenticate with the Device.

4575 Pursuant to the requirements in the SMETS and the CHTS requirement, Devices shall only
 4576 communicate at an application layer with other Devices that are currently in their Device Log
 4577 and are permitted by Device Based Access Controls (DBAC) as defined at Section 13.7.3.
 4578 Such communications shall always be secured using the shared secrets established
 4579 pursuant to Sections 13.7.4.

4580 Application layer communications within the scope of the DBAC requirement are HAN Only
 4581 Messages, including provision of information to a PPMID or Type 2 Device. Note that HAN
 4582 Only Messages between a PPMID and GSME have a structure that is specified in this GBCS
 4583 in the corresponding Use Cases, and those relate only to Add Credit and Activate
 4584 Emergency Credit Commands and the corresponding Responses.

4585 Each entry in a non CHF Device Log shall contain the Entity Identifier of the Authorised
 4586 Device and its deviceType.

4587 The Entity Identifier of a Device with DeviceType of
 4588 communicationsHubGasProxyFunction shall be the EUI 64-bit identifier of the ZigBee
 4589 radio interface installed in the Communications Hub.

4590 13.7.3 Device Based Access Control

4591 In relation to information and functionality within SMETS, a Device shall, when it is a
 4592 recipient of a Command or request for information from another Device on its SMHAN, only
 4593 attempt to action that Command when:

- 4594 • the sending Device's Entity Identifier is in the recipient Device's Device Log;
- 4595 • the ZSE cryptographic protection on the Message is authenticated using the Shared
 4596 Secret / Shared Secret Key established with the sending Device; and
- 4597 • the Command or request for information is explicitly allowed by a cell in Tables 13.7.3a
 4598 and 13.7.3b, in terms of the DeviceType of the sending (client) and receiving (service)
 4599 Device. The receiving Device shall determine the sending Device's DeviceType by
 4600 reference to its Device Log entry for that sending Device.

4601 Where a Device is a recipient of a Command or request for information from another Device
 4602 on its SMHAN that does not meet the access requirements of this Section 13.7.3, it shall:

- 4603 • generate an entry in the Security Log recording failed Authentication;
- 4604 • discard the Command or request for information without execution and without sending
 4605 a Response; and
- 4606 • send an Alert notifying the failed Authentication, constructed as specified in Section
 4607 6.2.4.2, populated with the relevant Alert Code from Section 16, to the Known Remote
 4608 Party specified in Table 16.2.

4609 An ESME shall not action any ZSE Local Change Supply command from a PPMID where
 4610 Proposed Supply Status has any value other than 0x02 ('Supply ON').

Device Name	Server / recipient	ESME	GSME	Comms Hub (GPF)	HCALC S	PPMID	Type 2 (IHD or CAD)
Client / sender	DeviceType	0	1	3	4	5	6
ESME	0	-	-	-	5.6.4.1 5.6.4.2	-	-
GSME	1	-	-	-	-	-	-

Comms Hub (GPF)	3	-	Request for Informatio n	-	-	-	-
HCALCS	4	8.5.2.1	-	-	-	-	-
PPMID	5	7.5.5.1 7.5.5.2 7.5.5.3 Request for Informatio n	7.5.4.1 7.5.4.2	Request for Informatio n	-	-	-
Type 2 (IHD or CAD)	6	Request for Informatio n	-	Request for Informatio n	-	-	-

Table 13.7.3a: Permitted Access by DeviceType, with Commands shown by SMETS reference

SMETS Ref	SMETS Command Name	ZSE Ref
5.6.4.1	Cancel Control HAN Connected Auxiliary Load Control Switch	Cancel Load Control Event
5.6.4.2	Request a HAN Connected Auxiliary Load Control Switch State Change	Load Control Event
8.5.2.1	Request Control of a HAN Connected Auxiliary Load Control Switch	Get Scheduled Events
7.5.5.1	Request Emergency Credit Activation	Select Available Emergency Credit
7.5.5.2	Request to Add Credit	Consumer Top Up
7.5.5.3	Request to Enable ESME Supply	Local Change Supply
7.5.4.1	Request Emergency Credit Activation	Select Available Emergency Credit
7.5.4.2	Request to Add Credit	Consumer Top Up

Table 13.7.3b: Mapping of Table 13.7.3a command references to SMETS names and ZSE

13.7.4 Use Case Requirements

This Section 13.7.4 details requirements which shall be complied with for all Join or Unjoin related Use Cases.

13.7.4.1 Use Cases covered

The types of Join Device related Messages, the Grouping names used in this Section 13.7.4, the associated Message Category and the valid recipient deviceType for each shall be as specified in Table 13.7.4.1³⁴. The SEC User Gateway Services Schedule (Service Request) Reference for all Join Use Cases shall be 8.7, and for Unjoin Use Cases shall be 8.8.

Message Code	Use Case Name	Valid recipient deviceType	Grouping	Message Category	Valid Business Originator role(s) for Command

³⁴ To avoid duplication of specification, the Use Cases here are grouped together, and the standard Use Case cross reference table is not used.

					invocation
0x000D	CS03A1 Method A Join (Meter)	eSME	Join Device	SME.C.C	Supplier
0x00AB	CS03A2 Method A Join (non Meter)	type1HANConnectedAuxiliaryLoadControlSwitch type1PrepaymentInterfaceDevice	Join Device	SME.C.C	Supplier
0x000E	CS03B Method B Join	gSME eSME communicationsHubGasProxyFunction	Join Device	SME.C.N C	Supplier, Access Control Broker
0x00AF	CS03C Method C Join	gSME type1PrepaymentInterfaceDevice	Join Device	SME.C.C	Supplier
0x000F	CS04AC Method A or C Unjoin	gSME eSME communicationsHubGasProxyFunction type1HANConnectedAuxiliaryLoadControlSwitch type1PrepaymentInterfaceDevice	Unjoin Device	SME.C.C	Supplier
0x0010	CS04B Method B Unjoin	gSME eSME communicationsHubGasProxyFunction	Unjoin Device	SME.C.N C	Supplier, Access Control Broker
0x0013	CS07 Read Device Join Details	gSME eSME communicationsHubGasProxyFunction type1HANConnectedAuxiliaryLoadControlSwitch type1PrepaymentInterfaceDevice		SME.C.N C	Supplier, Access Control Broker

4621 Table 13.7.4.1: Join Device related Commands, Grouping and Message Categories

4622 **13.7.4.2 Join Device Command and Response Processing**4623 **13.7.4.2.1 Construction of Commands**4624 ‘Join Device’ Command Payloads shall be constructed as specified in Section 13.7.4.5.2 and
4625 Cryptographic Protection I and Cryptographic Protection II shall be applied as required for a
4626 Command of the relevant Message Category.4627 For a Command (1) which complies with either Use Case ‘CS03A2 Method A Join (non
4628 Meter)’ or Use Case ‘CS03C Method C Join’ and (2) where the Device to which it is
4629 addressed has a deviceType equal to type1PrepaymentInterfaceDevice, the
4630 Access Control Broker’s Digital Signing Private Key shall be used in generating the KRP
4631 Signature.4632 **13.7.4.2.2 Device processing of Commands and Response handling**4633 The Device receiving a ‘Join Device’ Command shall undertake processing steps in the
4634 sequence defined in this Section 13.7.4.2.2. Should a step after step 1 be unsuccessful, the
4635 Device shall create a Response according to the requirements of Section 13.4.7, apply the
4636 Response Cryptographic Protection required for a Response of the relevant Message
4637 Category, and send the Response and shall not undertake any further steps defined in this
4638 Section 13.7.4.2.2.

4639 In processing a ‘Join Device’ Command, the Device shall:

4640 93. undertake Command Authenticity and Integrity Verification as required for a Command of
4641 this Message Category. The Security Credentials used to verify Cryptographic
4642 Protection 1 shall be:

- 4643 ○ those held in the {accessControlBroker, digitalSignature, management}
4644 Trust Anchor Cell, if deviceType equals type1PrepaymentInterfaceDevice;
4645 or
4646 ○ those held in the {supplier, digitalSignature, management} Trust Anchor
4647 Cell, if deviceType does not equal type1PrepaymentInterfaceDevice;
- 4648 94. verify the joinMethodAndRole as specified in Section 13.7.4.5.3;
- 4649 95. add the otherDeviceEntityIdentifier and otherDeviceType to its Device Log
4650 as specified in Section 13.7.4.5.4;
- 4651 96. if deviceType is eSME then undertake Key Establishment with the other Device as
4652 specified in Section 13.7.4.5.5;
- 4653 97. if joinMethodAndRole is methodC, and so the join is between a gSME and a
4654 type1PrepaymentInterfaceDevice, check that otherDeviceCertificate is
4655 present and validly structured. If the check succeeds the Device shall store, linked to
4656 this Device Log entry, details relating to otherDeviceCertificate, such that the
4657 Device is able to use subsequently the Shared Secret derived from
4658 otherDeviceCertificate and its own Private Key Agreement Key. If this check
4659 fails the Device shall set joinResponseCode to invalidOrMissingCertificate
4660 and processing shall be unsuccessful;
- 4661 98. set joinResponseCode to success, create a Response according to the
4662 requirements of Section 13.4.7, apply the Response Cryptographic Protection required
4663 for a Response of the relevant Message Category, and send the Response.

4664 *13.7.4.2.3 Response Processing*

4665 Response Recipient Verification may be undertaken as specified in this GBCS for a
4666 Response of the relevant Message Category. The joinResponseCode field in the
4667 Response may be decoded according to the ASN.1 definitions at Section 13.7.4.5.1.

4668 *13.7.4.3 ‘Unjoin Device’ Command and Response Processing*

4669 *13.7.4.3.1 Construction of Commands*

4670 ‘Unjoin Device’ Command Payloads shall be constructed as specified in Section 13.7.4.6.2
4671 and Cryptographic Protection I and Cryptographic Protection II shall be applied as required
4672 for a Command of the relevant Message Category.

4673 For a Command where the Device to which it is addressed has a deviceType equal to
4674 type1PrepaymentInterfaceDevice, the Access Control Broker’s Digital Signing Private
4675 Key shall be used in generating the KRP Signature.

4676 *13.7.4.3.2 Device processing of Commands and Response handling*

4677 The Device receiving an ‘Unjoin Device’ Command shall undertake processing steps in the
4678 sequence defined in this Section 13.7.4.3.2.

4679 In processing an ‘Unjoin Device’ Command, the Device shall:

4680 99. undertake Command Authenticity and Integrity Verification as required for a Command of
4681 this Message Category. The Security Credentials used to verify Cryptographic
4682 Protection 1 shall be:

- 4683 ○ those held in the {accessControlBroker, digitalSignature, management}
4684 Trust Anchor Cell, if deviceType equals type1PrepaymentInterfaceDevice;
4685 or

- 4686 ○ those held in the {supplier, digitalSignature, management} Trust Anchor
4687 Cell, if deviceType does not equal type1PrepaymentInterfaceDevice;
- 4688 100. set unjoinResponseCode to success;
- 4689 101. verify the otherDeviceEntityIdentifier matches an Entity Identifier currently
4690 recorded in its Device Log. If it does not then set unjoinResponseCode to
4691 otherDeviceNotInDeviceLog and process from step 5; otherwise process from step
4692 4;
- 4693 102. delete all information from the entry in its Device Log that has the same Entity
4694 Identifier as otherDeviceEntityIdentifier along with all shared cryptographic
4695 material related to that entry. If the deletion does not succeed, set
4696 unjoinResponseCode to otherFailure; and
- 4697 103. Create a Response according to the requirements of Section 13.7.4.6.2, apply the
4698 Response Cryptographic Protection required for a Response of the relevant Message
4699 Category, and send the Response.

4700 *13.7.4.3.3 Response Processing*

4701 Response Recipient Verification may be undertaken as specified in this GBCS for a
4702 Response of the relevant Message Category. The unjoinResponseCode field in the
4703 Response may be decoded according to the ASN.1 definitions at Section 13.7.4.6.1.

4704 *13.7.4.4 ‘CS07 Read Device Join Details’ Command and Response Processing*

4705 *13.7.4.4.1 Construction of Commands*

4706 ‘CS07 Read Device Join Details’ Command Payloads shall be constructed as specified in
4707 Section 13.7.4.7 and Cryptographic Protection II shall be applied as required for a Command
4708 of the SME.C.NC Message Category.

4709 *13.7.4.4.2 Device processing of Commands and Response handling*

4710 The Device receiving a ‘CS07 Read Device Join Details’ Command shall undertake
4711 processing steps in the sequence defined in this Section 13.7.4.4.2.

4712 In processing a ‘CS07 Read Device Join Details’ Command, the Device shall:

- 4713 104. undertake Command Authenticity and Integrity Verification as required for a
4714 Command of the SME.C.NC Message Category;
- 4715 105. set readLogResponseCode to success;
- 4716 106. attempt to read the Entity Identifier and deviceType for each of the entries in its
4717 Device Log. If the reading does not succeed for all entries, set readLogResponseCode
4718 to readFailure; otherwise populate deviceLogEntries using the data read from its
4719 Device Log; and

- 4720 107. create a Response according to the requirements of Section 13.7.4.7, apply the
4721 Response Cryptographic Protection required for a Response of the SME.C.NC Message
4722 Category, and send the Response.

4723 *13.7.4.4.3 Response Processing*

4724 Response Recipient Verification may be undertaken as specified in this GBCS for a
4725 Response of the SME.C.NC Message Category. The readLogResponseCode and
4726 deviceLogEntries fields in the Response may be decoded according to the ASN.1
4727 definitions at Section 13.7.4.7.

4728 **13.7.4.5 Component Requirements – Join**

4729 **13.7.4.5.1 Join Command and Response payloads – structure definition**

4730 Each instance of @JoinDevice.CommandPayload and of
 4731 @JoinDevice.ResponsePayload shall be an octet string containing the DER encoding of
 4732 the populated structure defined in this Section 13.7.4.5.1 which specifies the structure in
 4733 ASN.1 notation.

```

4734 JoinDevice DEFINITIONS ::= BEGIN
4735   CommandPayload ::=                               SEQUENCE
4736   {
4737     -- specify which type of joining is being authorised and,
4738     -- for Method A Joins, the role the Device is to play
4739
4740     joinMethodAndRole                         JoinMethodAndRole,
4741
4742     -- specify the Entity Identifier of the Device which is to be Joined with
4743
4744     otherDeviceEntityIdentifier                OCTET STRING,
4745
4746     -- specify the DeviceType of that other Device
4747
4748     otherDeviceType                           DeviceType,
4749
4750     -- provide the other Device's Key Agreement certificate, if and only if this
4751     -- is a join between a gSME and a type1PrepaymentInterfaceDevice.
4752     -- Certificate shall be as defined in IETF RFC 5912
4753
4754     otherDeviceCertificate                    Certificate OPTIONAL
4755
4756   }
4757
4758   -- detail whether the Command successful executed or, if it didn't,
4759   -- what the failure reason was
4760
4761   ResponsePayload ::=                         JoinResponseCode
4762
4763   JoinMethodAndRole ::=                      INTEGER
4764   {
4765     -- methodB is to be used where the other Device is a Type 2 Device or GPF.
4766     -- methodC is used where the Devices involved are a GSME and a PPMID.
4767     -- methodA is used otherwise.
4768     -- methodAInitiator is used where the Device this Command is targeted at
4769     -- should initiate the Key Agreement process
4770     -- methodAResponder is used where the Device this Command is targeted at
4771     -- should respond in the Key Agreement process, but shall not initiate it
4772
4773     methodAInitiator                         (0),
4774     methodAResponder                         (1),
4775     methodB                                (2),
4776     methodC                                (3)
4777   }
4778
4779   DeviceType ::=                            INTEGER
4780   {
4781     gSME                                  (0),
4782     eSME                                  (1),
4783     communicationsHubCommunicationsHubFunction (2),
4784     communicationsHubGasProxyFunction      (3),
4785     type1HANConnectedAuxiliaryLoadControlSwitch (4),
4786     type1PrepaymentInterfaceDevice        (5),
4787     type2                                (6)
4788   }
4789
4790   JoinResponseCode ::=                     INTEGER

```

```
4791 {  
4792     success          (0),  
4793     invalidMessageCodeForJoinMethodAndRole (1),  
4794     invalidJoinMethodAndRole    (2),  
4795     incompatibleWithExistingEntry (3),  
4796     deviceLogFull        (4),  
4797     writeFailure        (5),  
4798     keyAgreementNoResources (6),  
4799     keyAgreementUnknownIssuer   (7),  
4800     keyAgreementUnsupportedSuite (8),  
4801     keyAgreementBadMessage    (9),  
4802     keyAgreementBadKeyConfirm (10),  
4803     invalidOrMissingCertificate (11)  
4804 }  
4805  
4806 END
```

4807 **13.7.4.5.2 Constructing the @JoinDevice.CommandPayload and of @JoinDevice.ResponsePayload**

4808 @JoinDevice.CommandPayload shall have the structure defined in Section 13.7.4.5.1, and the Remote Party constructing the Command
 4809 shall populate with values according to Table 13.7.4.5.2a.

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
@JoinDevice.CommandPayload	SEQUENCE			
joinMethodAndRole	INTEGER	methodAInitiator (0), methodAResponder (1), methodB (2), methodC (3)		See Section 13.7.4.5.3 for valid values
otherDeviceEntityIdentifier	OCTET STRING	Entity Identifier	Mandatory	The Entity Identifier of the Device which is to be entered in this Device's Device Log
otherDeviceType	INTEGER	gSME (0), eSME (1), communicationsHubCommunicationsHubFunction (2), communicationsHubGasProxyFunction (3), type1HANConnectedAuxiliaryLoadControlSwitch (4), type1PrepaymentInterfaceDevice (5), type2 (6)	Mandatory	The DeviceType of the Device which is to be entered in this Device's Device Log
otherDeviceCertificate	Certificate	The Key Agreement Certificate currently in use by the other Device.	OPTIONAL	The other Device's Key Agreement certificate, which shall only be present if and only if this is a join between a gSME and a type1PrepaymentInterfaceDevice. Certificate shall be as defined in IETF RFC 5912.

4810 Table 13.7.4.5.2a: @JoinDevice.CommandPayload population @JoinDevice.ResponsePayload shall have the structure defined in Section

4811 13.7.4.5.1, and the Remote Party constructing the Command shall populate with values according to Table 13.7.4.5.2b.

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
@JoinDevice.ResponsePayload				
JoinResponseCode	INTEGER	Shall be populated according to the processing defined in Section 13.7.4.2.2	Mandatory	

4812 Table 13.7.4.5.2b: @JoinDevice.ResponsePayload population

4813 *13.7.4.5.3 Verification of joinMethodAndRole*

4814 The Device shall first verify the joinMethodAndRole specified in the Command Payload against the Message Code specified in the Grouping
 4815 Header of the Command according to Table 13.7.4.5.3a. If the check fails JoinResponseCode in the Response shall be set to the value
 4816 invalidMessageCodeForJoinMethodAndRole and no further verification checks in this Section 13.7.4.5.3a shall be undertaken.

Message Code	Use Case Name	Valid joinMethodAndRole
0x000D	CS03A1 Method A Join (Meter)	methodAInitiator
0x00AB	CS03B Method A Join (non Meter)	methodAResponder
0x000E	CS03A2 Method B Join	methodB
0x00AF	CS03C Method C Join	methodC

4817 Table 13.7.4.5.3a: Valid deviceMethod and joinMethodAndRole against Message Code The Device receiving a Join Device Command shall
 4818 verify joinMethodAndRole against its own DeviceMethod and the DeviceType specified in the otherDeviceType parameter of the
 4819 Command according to the requirements of the remainder of this Section 13.7.4.5.3.

4820 If joinMethodAndRole is methodB then the Device's verification of joinMethodAndRole shall be successful if there is a cell identified by
 4821 its own DeviceMethod, and the value of otherDeviceType (as identified in the Command) of a type defined in Table 13.7.4.5.3b, and that
 4822 cell contains 'success'. Otherwise, the verification shall fail and JoinResponseCode in the Response shall be set to the value
 4823 invalidJoinMethodAndRole.

4824

	otherDeviceType			
	communicationsHub GasProxyFunction	type1PrepaymentInterfaceDevice	type2	
DeviceType of Device to which the Command is addressed				
gSME	Success	-	-	
eSME	-	-		Success
communicationsHubGasProxyFunction	-	Success		Success

4825 Table 13.7.4.5.3b: joinMethodAndRole is methodB

4826 If joinMethodAndRole is methodAInitiator then the Device's verification of joinMethodAndRole shall be successful if there is a cell
 4827 identified by its own DeviceType, and the value of otherDeviceType (as identified in the Command) of a type defined in Table 13.7.4.5.3c,
 4828 and that cell contains 'success'. Otherwise, the verification shall fail and JoinResponseCode in the Response shall be set to the value
 4829 invalidJoinMethodAndRole.

	otherDeviceType		
	type1HANConnected AuxiliaryLoadControlSwitch	type1Prepayment InterfaceDevice	
DeviceType of Device to which the Command is addressed			
eSME	Success		Success

4825 Table 13.7.4.5.3c: joinMethodAndRole is methodB

4830 If joinMethodAndRole is methodAResponder then the Device's verification of joinMethodAndRole shall be successful if there is a cell
 4831 identified by its own DeviceType, and the value of otherDeviceType (as identified in the Command) of a type defined in Table 13.7.4.5.3d,
 4832 and that cell contains 'success'. Otherwise, the verification shall fail and JoinResponseCode in the Response shall be set to the value
 4833 invalidJoinMethodAndRole.

4834

	otherDeviceType
	eSME
DeviceType of Device to which the Command is addressed	
type1HANConnectedAuxiliaryLoadControlSwitch	Success
type1PrepaymentInterfaceDevice	Success

4835 Table 13.7.4.5.3d: joinMethodAndRole is methodAResponder If joinMethodAndRole is methodC then the Device's verification of
 4836 joinMethodAndRole shall be successful if there is a cell identified by its own DeviceType and the value of otherDeviceType (as
 4837 identified in the Command) in Table 13.7.4.5.3e and that cell contains 'success'. Otherwise, the verification shall fail and
 4838 JoinResponseCode in the Response shall be set to the value invalidJoinMethodAndRole.

	otherDeviceType	
	type1PrepaymentInterfaceDevice	gSME
DeviceType of Device to which the Command is addressed		
gSME	Success	-
type1PrepaymentInterfaceDevice	-	Success

4839 Table 13.7.4.5.3e: joinMethodAndRole is methodB
13.7.4.5.4 Adding the otherDeviceEntityIdentifier and otherDeviceType to the Device Log

4840 The Device shall undertake the following steps in the sequence specified:

4841 108. if the otherDeviceEntityIdentifier matches an Entity Identifier currently recorded in its Device Log, then the Device shall
 4842 compare deviceType in that log entry with otherDeviceType. If the Device types match then the addition is successful and processing
 4843 within this Section 13.7.4.5.4 shall cease; otherwise the Device shall set joinResponseCode to incompatibleWithExistingEntry
 4844 and processing within this Section 13.7.4.5.4 shall cease;

- 4845 109. the Device shall check if there is capacity for an additional entry in its Device Log. If there is not, the Device shall set
 4846 joinResponseCode to deviceLogFull and processing within this Section 13.7.4.5.4 shall cease; and
- 4847 110. the Device shall attempt to create a new Device Log entry using otherDeviceEntityIdentifier and otherDeviceType. If that
 4848 entry is not successfully created, the Device shall set joinResponseCode to writeFailure.

4849 ***13.7.4.5.5 Undertaking Key Establishment with the other Device***

4850 The Device shall initiate, and attempt to complete, Key Establishment according to the ZSE requirements. The initiating Device shall wait a
 4851 minimum of two seconds before timing out any key establishment operation.

4852 Should there be errors that result in that process not completing, the Device shall set joinResponseCode to the value specified by Table
 4853 13.7.4.5.5.

ZSE Response Code ³⁵	Value of joinResponseCode
NO_RESOURCES	keyAgreementNoResources
UNKNOWN_ISSUER	keyAgreementUnknownIssuer
UNSUPPORTED_SUITE	keyAgreementUnsupportedSuite
BAD_MESSAGE	keyAgreementBadMessage
BAD KEY_CONFIRM	keyAgreementBadKeyConfirm

4854 Table 13.7.4.5.5: joinResponseCode mapping to ZSE Responses

13.7.4.6 Component Requirements – Unjoin

4855 ***13.7.4.6.1 Unjoin Command and Response payloads – structure definition***

4856 Each instance of @UnjoinDevice.CommandPayload and of @UnjoinDevice.ResponsePayload shall be an octet string containing the
 4857 DER encoding of the populated structure defined in this Section 13.7.4.6.1 which specifies the structure in ASN.1 notation.

4858 UnjoinDevice DEFINITIONS ::= BEGIN
 4859
 4860 CommandPayload ::= OtherDeviceEntityIdentifier
 4861 -- specify the Entity Identifier of the Device for which authorisation
 4862 -- is to be removed

³⁵ As defined in the ZSE specification

```

4863
4864     OtherDeviceEntityIdentifier ::=          OCTET STRING
4865
4866     ResponsePayload ::=                      UnjoinResponseCode
4867
4868         -- detail whether the Command successful executed or, if it didn't,
4869         -- what the failure reason was
4870
4871     UnjoinResponseCode ::=                  INTEGER
4872     {
4873         success                           (0),
4874         otherDeviceNotInDeviceLog        (1),
4875         otherFailure                     (2)
4876     }
4877
4878 END

```

13.7.4.6.2 Constructing the @UnjoinDevice.CommandPayload and of @UnjoinDevice.ResponsePayload

@UnjoinDevice.CommandPayload shall have the structure defined in Section 13.7.4.6.1, and the Remote Party constructing the Command shall populate with values according to Table 13.7.4.6.2a.

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
@UnjoinDevice.CommandPayload				
OtherDeviceEntityIdentifier	OCTET STRING	Entity Identifier	Mandatory	The Entity Identifier of the Device which is to be removed from this Device's Device Log

Table 13.7.4.6.2a: @UnjoinDevice.CommandPayload population @UnjoinDevice.ResponsePayload shall have the structure defined in Section 13.7.4.6.1, and the Remote Party constructing the Command shall populate with values according to Table 13.7.4.6.2b.

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
@UnjoinDevice.ResponsePayload				
unjoinResponseCode	INTEGER	success (0),	Mandatory	Shall be populated according to the

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
		otherDeviceNotInDeviceLog (1), otherFailure (2)		processing defined in Section 13.7.4.3

4884 Table 13.7.4.6.2b: @UnjoinDevice.ResponsePayload population

4885 **13.7.4.7 CS07 Read Device Join Details Command and Response payloads – structure** 4886 **definition**

4887 Each instance of @ReadDeviceLog.CommandPayload and of
 4888 @ReadDeviceLog.ResponsePayload shall be an octet string containing the DER
 4889 encoding of the populated structure defined in this Section 13.7.4.7 which specifies the
 4890 structure in ASN.1 notation.

```
4891 ReadDeviceLog DEFINITIONS ::= BEGIN
4892 
4893   CommandPayload ::= NULL
4894 
4895   ResponsePayload ::= SEQUENCE
4896   {
4897     -- detail whether the Command successful
4898     readLogResponseCode           ReadLogResponseCode,
4899 
4900     -- if it was, return the Log Entries
4901     deviceLogEntries             SEQUENCE OF DeviceLogEntry OPTIONAL
4902   }
4903 
4904   DeviceLogEntry ::= SEQUENCE
4905   {
4906     deviceIdentifier            OCTET STRING,
4907     deviceType                  DeviceType
4908   }
4909 
4910   DeviceType ::= INTEGER
4911   {
4912     gSME                      (0),
4913     eSME                      (1),
4914     communicationsHubFunction (2),
4915     communicationsHubGasProxyFunction (3),
4916     type1HANConnectedAuxiliaryLoadControlSwitch (4),
4917     type1PrepaymentInterfaceDevice (5),
4918     type2                      (6)
4919   }
4920 
4921 
4922   ReadLogResponseCode ::= INTEGER
4923   {
4924     success                   (0),
4925     readFailure               (1)
4926   }
4927 
4928 
4929 END
```

4930 **13.8 GCS59 / 62 GPF Device Log Backup and Restore**

4931 **13.8.1 Introduction to GPF Device Log Backup and Restore -** **informative**

4933 **13.8.1.1 The role of pair-wise authorisation - informative**

4934 This Section 13.8 includes the Use Cases related to the backing up and restoring of the
 4935 GPF's Device Log. This is to cater for situation where the existing Communications Hub
 4936 fails and has to be replaced.

4937 In summary:

- 4938 • a GPF sends an Alert whenever its Device Log changes (unless the change is as a
 4939 result of a restore of the Device Log). That Alert contains the contents of the GPF's
 4940 Device Log after the change has been made; and

- 4941 • the Restore GPF Device Log Command shall contain the same structure of Device Log
 4942 contents. If successful, the Command will place those contents in to the GPF's Device
 4943 Log and will have triggered the processing required to authorise the specified Devices
 4944 application layer interaction with the GPF, where required.

4945 **13.8.1.2 The format of Message Payloads - informative**

4946 In common with other GBCS Remote Messages related to the management of Security
 4947 Credentials, the Payloads of Alerts, Commands and Responses defined in this Section 13.8
 4948 are specified using ASN.1, with DER encoding to be applied to Command and Response
 4949 payloads.

4950 Each entry in a GPF Device Log shall contain the Entity Identifier of the Authorised Device
 4951 and its deviceType.

4952 **13.8.2 GCS62 Backup GPF Device Log**

4953 **13.8.2.1 Description**

4954 This Section 13.8.2 covers the creation, validation and processing of Alerts resulting from
 4955 changes to the GPF Device Log. One such Alert shall be generated each time that the GPF
 4956 Device Log changes, except where the change arises from a GPF Device Log Restore
 4957 Command.

4958 **13.8.2.2 Use Case Cross References**

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Alert
Message Type Category	SME.A.NC
Capable of future dated invocation?	N/A
Protection Against Replay Required?	N/A
SEC User Gateway Services Schedule (Service Request) Reference	8.12
Valid Initiating Device(s)	GPF
Valid Business Target role(s) for Alert	Access Control Broker
Valid Response Recipient role(s) (only for Messages Authorised by the Access Control Broker on behalf of parties not known to the Device) [Remote Party Messages Only]	N/A
Valid initiating Device type(s) [HAN Only Messages]	N/A
Protocol	ASN.1

4959 Table 13.8.2.2: Use Case Cross References for GPF Device Log Backup Alert

4960 **13.8.2.3 Construction of Alerts**

4961 GPF Device Log Backup Alert Payloads shall be constructed according to the requirements
 4962 of Section 13.8.4.1 and populated as specified in Table 13.8.2.3.

4963 MAC Header, Grouping Header and SMD-KRP MAC shall be populated as required for an
 4964 Alert of the SME.A.NC Message Category, with the Message Code being 0x00B2. Note that
 4965 the Business Target ID in the Grouping Header shall always contain the Entity Identifier of
 4966 the Access Control Broker.

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
@GPFDeviceLog.BackupAlertPayload	SEQUENCE			
alertCode	INTEGER	0x0071	Mandatory	Fixed value specifying that this is a GPF Device Log Backup Alert
backupDateTime	GeneralizedTime	The date-time at which this Alert was created	Mandatory	This is based on the Device's own clock
deviceLogEntries	SEQUENCE OF		OPTIONAL	There may be 0, 1 or many entries in the Log. The following two fields will be repeated as many times as there are Device Log Entries
deviceEntityIdentifier	OCTET STRING	Entity Identifier	Mandatory	The Entity Identifier of the Device to which this entry relates.
deviceType	INTEGER	type1PrepaymentInterfaceDevice (5), type2 (6)	Mandatory	The DeviceType of the Device to which this entry relates. These are the only valid entries for the GPF Device Log.

Table 13.8.2.3: @GPFDeviceLog.BackupAlertPayload population [13.8.2.4 Processing of Alerts](#)

SMD-KRP MAC may be verified by the Access Control Broker as per Section 6.8.3.

13.8.3 GCS59 GPF Device Log Restore

13.8.3.1 Description

This section covers the creation, validation and processing of Commands to restore the GPF Device Log, and the creation and validation of the corresponding Response.

13.8.3.2 Use Case Cross References

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Command and Response

Cross Reference	Value
Message Type Category	SME.C.NC
Capable of future dated invocation?	No
Protection Against Replay Required?	Yes
SEC User Gateway Services Schedule (Service Request) Reference	8.12
Valid Target Device(s)	GPF
Valid Business Originator role(s) for Command	Access Control Broker
Valid Response Recipient role(s) (only for Messages Authorised by the Access Control Broker on behalf of parties not known to the Device) [Remote Party Messages Only]	N/A
Valid initiating Device type(s) [HAN Only Messages]	N/A
Protocol	ASN.1

4974 Table 13.8.3.2: Use Case Cross References for GPF Device Log Restore

13.8.3.3 Construction of Command

4975 GPF Device Log Restore Command Payloads shall be constructed according to the requirements of Section 13.8.4.1 and populated as
 4976 specified in Table 13.8.3.3.

4977 MAC Header, Grouping Header, KRP Signature and ACB-SMD MAC shall be populated as required for a Command of the SME.C.C Message
 4978 Category.

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
@GPFDeviceLog.RestoreComm andPayload	SEQUENCE			
deviceLogEntries	SEQUENCE OF		OPTIONAL	There may be 0, 1 or many entries in the Log. The following two fields will be repeated as many times as there are Device Log Entries. Note that there would be no effect if the Command had no deviceLogEntries.

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
deviceEntityIdentifier	OCTET STRING	Entity Identifier	Mandatory as part of each entry that is present	The Entity Identifier of the Device to which this entry relates.
deviceType	INTEGER	type1PrepaymentInterfaceDevice (5), type2 (6)	Mandatory as part of each entry that is present	The DeviceType of the Device to which this entry relates. These are the only valid entries for the GPF Device Log. Note that the GSME does not need to be in the GPF's Device Log, since the GPF only receives information from the GSME.

4979 Table 13.8.3.3: @GPFDeviceLog.RestoreCommandPayload population 13.8.3.4 Device processing of Command and Response handling

4980 The GPF receiving a GPF Device Log Restore Command shall undertake processing steps in the sequence defined in this Section 13.8.3.4.
 4981 The Device shall undertake Command Authenticity and Integrity Verification as required for a Command of this Message Category, and then, if
 4982 successful, for each DeviceLogEntry in deviceLogEntries, shall:
 4983 111. set deviceLogEntry in the corresponding ResponseOutcome to the values of this DeviceLogEntry in deviceLogEntries;
 4984 112. set joinResponseCode in the corresponding ResponseOutcome to success;
 4985 113. if the deviceEntityIdentifier matches an Entity Identifier currently recorded in its Device Log, compare deviceType in that log
 entry with otherDeviceType. If the Device types match then the addition is successful and processing of this DeviceLogEntry shall
 cease; otherwise the Device shall set joinResponseCode to incompatibleWithExistingEntry and processing of this
 DeviceLogEntry shall cease;
 4986 114. check if there is capacity for an additional entry in its Device Log. If there is not, the Device shall set joinResponseCode to
 deviceLogFull and processing of this DeviceLogEntry shall cease; and
 4987 115. attempt to create a new Device Log entry using deviceEntityIdentifier and deviceType. If that entry is not successfully
 created, the Device shall set joinResponseCode to writeFailure and processing of this DeviceLogEntry shall cease.
 4988 Once all DeviceLogEntry in deviceLogEntries have been processed, the GPF shall populate the Response Payload according to the
 4989 requirements of Section 13.8.4.1 using the ResponseOutcomes produced by the processing in this Section 13.8.3.4, construct MAC Header,
 4990 Grouping Header and apply the Response Cryptographic Protection required for a Response of the SME.C.NC Message Category, and send
 4991 the Response.

4997 13.8.4 Common Requirements

4998 **13.8.4.1 GPF Device Log Backup Alert, Restore Command and Restore Response Payloads – structure definition**

4999 Each instance of `@GPFDeviceLog.BackupAlertPayload`, `@GPFDeviceLog.RestoreCommandPayload` and of
5000 `@GPFDeviceLog.RestoreResponsePayload` shall be an octet string containing the DER encoding of the populated structure defined in this
5001 Section 13.8.4.1 which specifies the structure in ASN.1 notation.

```
5002 GPFDeviceLog DEFINITIONS ::= BEGIN
5003
5004 BackupAlertPayload ::= SEQUENCE
5005 {
5006     -- specify the Alert Code
5007     alertCode           INTEGER(0..4294967295),
5008
5009     -- specify the date-time of the backup
5010     backupDateTime       GeneralizedTime,
5011
5012     -- detail the entries in the Device Log now that the change has been made
5013     deviceLogEntries    SEQUENCE OF DeviceLogEntry
5014
5015 }
5016
5017 RestoreCommandPayload ::= SEQUENCE
5018 {
5019     -- list the Device Log entries that are to be added
5020     deviceLogEntries    SEQUENCE OF DeviceLogEntry
5021
5022 }
5023
5024 DeviceLogEntry ::= SEQUENCE
5025 {
5026     -- specify the Entity Identifier of the Device
5027     deviceEntityIdentifier OCTET STRING,
5028
5029     -- specify the DeviceType of that Device
5030     deviceType            DeviceType
5031
5032 }
5033
5034 RestoreResponsePayload ::= SEQUENCE
```

```
5036
5037 {
5038     -- for each DeviceLog Entry, detail whether the Command successfully executed or, if it didn't, what the failure reason was
5039
5040     restoreOutcomes
5041         SEQUENCE OF RestoreOutcome
5042
5043     RestoreOutcome ::= SEQUENCE
5044     {
5045         deviceLogEntry
5046         JoinResponseCode
5047     }
5048
5049     DeviceType ::= INTEGER
5050     {
5051         gSME
5052         eSME
5053         communicationsHubCommunicationsHubFunction
5054         communicationsHubGasProxyFunction
5055         type1HANConnectedAuxiliaryLoadControlSwitch
5056         type1PrepaymentInterfaceDevice
5057         type2
5058     }
5059
5060     JoinResponseCode ::= INTEGER
5061     {
5062         success
5063         invalidMessageCodeForJoinMethodAndRole
5064         invalidJoinMethodAndRole
5065         incompatibleWithExistingEntry
5066         deviceLogFull
5067         writeFailure
5068         keyAgreementNoResources
5069         keyAgreementUnknownIssuer
5070         keyAgreementUnsupportedSuite
5071         keyAgreementBadMessage
5072         keyAgreementBadKeyConfirm
5073         invalidOrMissingCertificate
5074     }
5075 END
```

14 Apply Prepayment Top Up to an ESME or GSME

14.1 Defined Terms

The following terms used in this Section 14 shall have the meanings defined in this Table 14.1.

Defined Term	Meaning
Currency Unit	Shall be either GB Pound or European Central Bank Euro
Maximum Credit Threshold	Shall be the maximum value of any single Pre-payment Top Up. Its value shall be interpreted by the Device in Currency Units (whole currency units only)
Maximum Meter Balance Threshold	Shall be the maximum total credit value recorded on the ESME / GSME. Its value shall be interpreted by the Device in Currency Units (whole currency units only)
Highest UTRN Counter	The highest numerical value of any UTRN Counter in the UTRN Counter Cache
Prepayment Token Decimal (PPTD)	Shall have the meaning specified in Section 14.3.1
Prepayment Top Up Token (PTUT)	Shall have the meaning specified in Section 14.3.2
Unique Transaction Reference Number (UTRN)	Shall have the meaning specified in Section 14.3.3
UTRN Check Digit	Shall be the 20 th digit of the UTRN
UTRN Counter Cache	Shall be an array of 100 entries, each entry containing an unsigned integer of 32 bits in length and an associated flag to indicate whether the UTRN Counter represented by the integer relates to a locally entered Prepayment Top Up, a network delivered Prepayment Top Up or has been set as a floor value on execution of an Update Security Credentials Command. The array shall be arranged as a circular buffer such that, when full, further writes shall cause the lowest numerical value entry to be overwritten
UTRN Counter	The 32 most significant bits of the Originator Counter

Table 14.1: Meanings of Defined Terms

14.2 Description - informative

This section covers the application of a Prepayment Top Up, that has been purchased for a particular ESME or GSME, to that ESME or GSME.

It covers four options:

- applying a Prepayment Top Up to an ESME without consumer intervention;
- applying a Prepayment Top Up to a GSME without consumer intervention;
- applying a Prepayment Top Up to an ESME or GSME with consumer entry of a numeric code on the ESME or GSME; and

- 5089 • applying a Prepayment Top Up to an ESME or GSME with consumer entry of a numeric
 5090 code on a PPMID.

5091 Some requirements are common to all four options. Accordingly, this Section 14 is split in to
 5092 five subsections:

- 5093 • an initial subsection covering requirements common to all four options; and
 5094 • four subsequent subsections covering one option in each subsection.

5095 By way of context:

- 5096 • any Prepayment Top Up Message is a Remote Party Command in GBCS terms
 5097 (because it is from a Remote Party to a GSME or ESME). The means of delivery
 5098 (typing in on meter, typing in on PPMID, sending over WAN, etc.) does not affect this
 5099 classification;
- 5100 • as a Remote Party Command, it must result in the GSME or ESME generating a
 5101 Response back to the Remote Party who issued it (so the Supplier), unless there is an
 5102 Authentication failure (in which case the Supplier has to be sent an Alert), as per
 5103 SMETS and CHTS;
- 5104 • because the ranges are exclusive, the Originator Counter in Prepayment Top Up
 5105 transactions cannot collide with the Originator Counter in any other transaction; and
- 5106 • there is no requirement to include the Device's ID explicitly in the locally entered
 5107 transaction, so a PPMID joined to more than one Smart Meter will need to allow the
 5108 Consumer to pick which Smart Meter the Prepayment Top Up is for.

5109 **14.3 Common Requirements**

5110 **14.3.1 Construction of the PPTD**

5111 The PPTD shall be a 19 decimal digit integer. The most significant two digits of the PPTD
 5112 shall always be between 73 and 96, which shall be constructed and represented according
 5113 to the requirements of this Section 14.3.1.

5114 The decimal representation of the PPTD shall be the result of the addition of
 5115 7,394,156,990,786,306,048 to the decimal representation of the PTUT.

5116 **14.3.2 Construction of the PTUT**

5117 The PTUT shall be an unsigned 64 bit integer (so 8 octets), which shall be constructed and
 5118 represented according to the requirements of this Section 14.3.2.

5119 The bits within the PTUT shall be numbered from 63 for the most significant bit through to 0
 5120 for the least significant bit.

5121 The bits of the PTUT shall be set to the values in Table 14.3.2.

PTUT component	Value	Bits	Note
PTUT Lead	0b000	63-61	Fixed Value
PTUT Sub Class	0b0000	60-57	Fixed value
PTUT Value Class	0b00 if PTUT Value is to be interpreted as multiples of 1/100 of Currency Unit; OR 0b01 if PTUT Value is to be	56-55	If Currency Unit is set to GB Pounds on the ESME or GSME, 0b00 means PTUT Value will be interpreted as GB Pennies; and 0b01 means PTUT Value will be interpreted as GB Pounds.

PTUT component	Value	Bits	Note
	interpreted as multiples of Currency Unit.		
PTUT Value	The quantum of the PTUT expressed as an unsigned binary number of 13 bits in length, so with leading binary zeros where required.	54-42	Thus, the maximum value is either: £81.91 if PTUT Value Class =0b00; or £8,191.00 if PTUT Value Class =0b01.
PTUT Truncated Originator Counter	Bits 41-32 of the Originator Counter	41-32	Used for Protection Against Replay purposes when the transaction is entered locally.
PTUT Supplier MAC	See Section 14.3.4	31-0	

5122 Table 14.3.2: Values of PTUT bits

5123 14.3.3 Construction of the Unique Transaction Reference Number (UTRN)

5124

5125 The Unique Transaction Reference Number (UTRN) shall be a 20 decimal digit which shall
 5126 be the 19 decimal digits of the PPTD with a 20th decimal digit which shall be appended after
 5127 the least significant digit of the 19 decimal digit representation of PPTD. This 20th decimal
 5128 digit shall be the UTRN Check Digit. The UTRN Check Digit shall be calculated according to
 5129 the requirements of Section 14.8.

5130 14.3.4 Construction of the PTUT Supplier MAC

5131 The PTUT Supplier MAC shall only be calculated once the 32 most significant bits of PTUT
 5132 (bits 63-32 of the PTUT) have been populated as per the requirements of Section 14.3.2.

5133 The Remote Party, whose Security Credentials are stored against the Supplier role of the
 5134 target Device, shall calculate a MAC using the parameters in Table 14.3.4 then setting the
 5135 PTUT Supplier MAC to be the 32 least significant bits of the 128 bit MAC produced by the
 5136 MAC calculation.

Input Parameter	Value	Note
To calculate the Shared Secret ('Z') input to the KDF:		
Private Key Agreement Key	Supplier's Prepayment Top Up Key Agreement Key [which the Supplier may elect to be different than the Key Agreement Key they use for other interactions with the Device]	
Public Key Agreement Key	Device's	
The other input to the KDF ('OtherInfo') shall be calculated according to the requirements of Section 4.3.3.3.		
As input to the GMAC function, the IV shall be constructed according to the requirements of Section 4.3.3.4, the Plaintext shall be empty and:		
Additional Authenticated Data shall be the concatenation:	0x11 Message Identifier 32 most significant bits of the PTUT	

5137 Table 14.3.4: Calculation of the PTUT Supplier MAC

5138 **14.3.5 Validating the PTUT Supplier MAC**

5139 To validate the PTUT Supplier MAC, the Device shall calculate the MAC using the
5140 parameters in Table 14.3.5, then ensure the 32 least significant bits of the 128 bit MAC
5141 produced by the MAC calculation has the same value as the PTUT Supplier MAC.

5142

Input Parameter	Value	Note
To calculate the Shared Secret ('Z') input to the KDF:		
Private Key Agreement Key	Device's	
Public Key Agreement Key	Supplier's Prepayment Top Up Key Agreement Key	
The other input to the KDF ('OtherInfo') shall be calculated according to the requirements of Section 4.3.3.3.		
As input to the GMAC function, the IV shall be constructed according to the requirements of Section 4.3.3.4, the Plaintext shall be empty and:		
Additional Authenticated Data shall be the concatenation:	0x11 Message Identifier 32 most significant bits of the PTUT	

Table 14.3.5: Validation of the PTUT Supplier MAC

5143 14.3.6 Checking the UTRN Counter against the UTRN Counter Cache

5144 The Device shall set the UTRN Counter to be the 32 most significant bits of the Originator Counter.

5145 The Device shall check that the UTRN Counter is strictly numerically greater than the numerically lowest value in the UTRN Counter Cache, and is not equal to any value in the UTRN Originator Counter Cache.

5150 14.3.7 Updating the UTRN Counter Cache

5151 Where the Prepayment Top Up is successfully applied and prior to sending any Response, the Device shall add a new entry to the UTRN Counter Cache whose UTRN Counter value shall be set to the 32 most significant bits of Originator Counter and whose flag shall be set to record this Prepayment Top Up either as a network delivered Prepayment Top Up or as a locally entered Prepayment Top Up, as appropriate.

5156 14.3.8 Validating the Maximum Credit Values

5157 14.3.8.1 Maximum Credit Threshold

5158 The Device shall ensure that the top-up value specified by PTUT Value Class and PTUT Value does not exceed the Device's Maximum Credit Threshold parameter.

5160 14.3.8.2 Maximum Meter Balance Threshold

5161 The Device shall ensure that the top-up value specified by PTUT Value Class and PTUT Value when added to the Device's Credit Balance does not exceed the Device's Maximum Meter Balance Threshold parameter.

5164 14.3.9 Validating the PTUT Sub-Class

5165 The Device shall ensure that the value specified by PTUT Sub-Class is of value 0b0000.

5166 14.4 CS01a Applying a Prepayment Top Up to an ESME 5167 without consumer intervention

5168 14.4.1 Description

5169 This section covers the application of a Prepayment Top Up that has been bought for an
5170 ESME to that ESME, in the case where the consumer does not enter any details on Devices
5171 in their premises.

5172 14.4.2 Use Case Cross References

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Command and Response
Message Type Category	SME.C.NC but with additional cryptographic processing specified in Sections 14.3.4 and 14.3.5
Capable of future dated invocation?	No
Protection Against Replay Required?	The Protection Against Replay mechanisms for Prepayment Top Ups are specified in Section 14.3.6. The Protection Against Replay mechanisms specified elsewhere in the GBCS do not apply
SEC User Gateway Services Schedule (Service Request) Reference	2.2
Valid Target Device(s)	ESME
Valid Business Originator role(s) for Command invocation (and so, for DLMS COSEM Commands, which Application Association is to be used for delivery of the Command to the ESME) [Remote Party Messages Only]	Supplier
Valid Response Recipient role(s) (only for Messages Authorised by the Access Control Broker on behalf of parties not known to the Device) [Remote Party Messages Only]	N/A
Valid initiating Device type(s) [HAN Only Messages]	N/A
Protocol	DLMS COSEM

Table 14.4.2: Use Case Cross References for Prepayment Top Up to an ESME without consumer intervention

5173 14.4.3 Pre-conditions

5174 None.

5175 14.4.4 Detailed Steps

5176 The Device shall undertake the checks set out in this Section 14.4.4 in the sequence laid out:

- 5177 • only once all checks in Section 6.2.4.1.1 have been successfully completed; and
- 5178 • before undertaking any other processing of the Command.

5179 If any of the checks specified in this Section 14.4.4 fail, the Device shall not carry out further
 5180 checks, and the requirements of Section 6.2.4.2 shall apply. Otherwise, processing shall
 5181 continue as per the requirements of Section 6.2.4.1.2. Where that check is successful,
 5182 processing shall continue as below.

5183 **14.4.4.1 Verifying against the maximum credit values**

5184 The Device shall carry out the checks specified in Section 14.3.8.1 and Section 14.3.8.2.

5185 **14.4.4.2 Verifying the Originator Counter**

5186 The Device shall verify the Originator Counter against the UTRN Counter Cache according
 5187 to Section 14.3.6.

5188 **14.4.4.3 Validating the PTUT Supplier MAC**

5189 The Device shall validate the PTUT Supplier MAC according to Section 14.3.5.

5190 **14.4.5 Response Construction**

5191 At the Response Construction stage, the Device shall first update the UTRN Counter Cache
 5192 according to Section 14.3.7, and shall then populate the Response according to the
 5193 requirements of the Message Template for CS01a.

5194 **14.5 CS01b Applying a Prepayment Top Up to a GSME 5195 without consumer intervention**

5196 **14.5.1 Description**

5197 This section covers the application of a Prepayment Top Up that has been bought for a
 5198 GSME to that GSME, in the case where the consumer does not enter any details on Devices
 5199 in their premises, except for the additional processing defined in this section.

5200 **14.5.2 Use Case Cross References**

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Command and Response
Message Type Category	See Table 14.4.2
Capable of future dated invocation?	No
Protection Against Replay Required?	See Table 14.4.2
SEC User Gateway Services Schedule (Service Request) Reference	2.2
Valid Target Device(s)	GSME
Valid Business Originator role(s) for Command invocation (and so, for DLMS COSEM Commands, which Application Association is to be used for delivery of the Command to the ESME) [Remote Party Messages Only]	Supplier
Valid Response Recipient role(s) (only for Messages Authorised by the Access Control Broker on behalf of parties not known to the Device) [Remote Party Messages Only]	N/A
Valid initiating Device type(s) [HAN Only Messages]	N/A

Cross Reference	Value
Protocol	GBZ

5201 Table 14.5.2: Use Case Cross References for Prepayment Top Up to a GSME without consumer
 5202 intervention

14.5.3 Pre-conditions

5204 None.

14.5.4 Detailed Steps

5206 The Device shall undertake the checks set out in this Section 14.5.4 in the sequence laid out:

- only once all checks in Section 6.2.4.1.1 have been successfully completed; and
- before undertaking any other processing of the Command.

14.5.4.1 Verifying against the maximum credit values

5210 The Device shall carry out the checks specified in Section 14.3.8.1 and Section 14.3.8.2.

14.5.4.2 Verifying the Originator Counter

5212 The Device shall verify the Originator Counter against the UTRN Counter Cache according
 5213 to Section 14.3.6.

14.5.4.3 Validating the PTUT Supplier MAC

5215 The Device shall validate the PTUT Supplier MAC according to Section 14.3.5.

5216 If any of the checks specified in this Section 14.5.4 fail, the requirements of Section 6.2.4.2
 5217 shall apply. Otherwise, processing shall continue as per the requirements of Section
 5218 6.2.4.1.2.

14.5.5 Response Construction

5220 At the Response Construction stage, the Device shall first update the UTRN Counter Cache
 5221 according to Section 14.3.7 and shall then populate the Response according to the
 5222 requirements of Use Case CS01b.

14.6 Applying a Prepayment Top Up to an ESME or GSME with consumer entry of a numeric code on the ESME or GSME

14.6.1 Description

5227 This section covers the application of a Prepayment Top Up that has been bought for a
 5228 GSME or ESME to that GSME or ESME in the case where the consumer enters the
 5229 corresponding UTRN on the GSME or ESME.

5230 The Use Case covering the Response is referenced in Section 14.6.5.

14.6.2 Use Case Cross References

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Command and Response
Message Type Category	This is a Variant Message type. The Command shall be the UTRN constructed in accordance with Section 14.3.3. The Command includes

Cross Reference	Value
	cryptographic protections as specified in Sections 14.3.4 and 14.3.5. The Response shall be of Message Type Category SME.C.NC. An Alert as specified in SMETS for locally entered commands is not required
Capable of future dated invocation?	No
Protection Against Replay Required?	See Table 14.4.2
SEC User Gateway Services Schedule (Service Request) Reference	N/A
Valid Target Device(s)	ESME or GSME
Valid Business Originator role(s) for Command invocation (and so, for DLMS COSEM Commands, which Application Association is to be used for delivery of the Command to the Device) [Remote Party Messages Only]	Supplier
Valid Response Recipient role(s) (only for Messages Authorised by the Access Control Broker on behalf of parties not known to the Device) [Remote Party Messages Only]	N/A
Valid initiating Device type(s) [HAN Only Messages]	N/A
Protocol	Outside of protocols since entered via User Interface

Table 14.6.2: Use Case Cross References for Prepayment Top Up through consumer UTRN entry

5232 **14.6.3 Pre-conditions**

5233 None.

5234 **14.6.4 Detailed Steps**

5235 **14.6.4.1 Detailed Steps/Sequence**

5236 The Device shall undertake the validation checks set out in this Section 14.6.4.1 before
 5237 undertaking any other processing of the Command. The validation checks shall be
 5238 undertaken in the sequence laid out. Should a validation check fail, subsequent validation
 5239 checks shall not be undertaken by the Device.

5240 Should any of the checks fail (save for the optional UTRN Check Digit verification), the
 5241 requirements of Section 6.2.4.2 shall apply.

5242 **14.6.4.1.1 Verifying the UTRN Check Digit**

5243 The Device:

- 5244 • may validate the 20th digit (the UTRN Check Digit) as specified at Section 14.8
 5245 (Calculating and Verifying the UTRN Check Digit); and
- 5246 • shall disregard the 20th decimal digit to determine PPTD prior to undertaking any
 5247 subsequent checks.

5248 **14.6.4.1.2 Using the PPTD to calculate the PTUT**

5249 PTUT shall take the value of PPTD minus 7,394,156,990,786,306,048.

5250 The Device shall interpret the resulting unsigned integer according to Table 14.6.4.1.2.

5251

PTUT component	Bits
PTUT Sub Class	60-57
PTUT Value Class	56-55
PTUT Value	54-42
PTUT Truncated Originator Counter	41-32
PTUT Supplier MAC	31-0

5252 Table 14.6.4.1.2: Interpretation of the PTUT [14.6.4.1.3 Verifying PTUT subclass category](#)

5253 The Device shall carry out the checks specified in Section 14.3.9.

14.6.4.1.4 Verifying against the maximum credit values

5255 The Device shall carry out the checks specified in Section 14.3.8.1 and Section 14.3.8.2.

14.6.4.1.5 Deriving the Originator Counter³⁶

5257 The Originator Counter shall be derived by:

- 5258 4. creating four 32 bit signed integer variables p, q, r and s;
- 5259 5. setting p = the numeric value of the 10 least significant bits of Highest UTRN Counter;
- 5260 6. setting q = (the numeric value Highest UTRN Counter) – p;
- 5261 7. setting r = the numeric value of PTUT Truncated Originator Counter;
- 5262 8. if r < (p – 2⁹) then setting s = (r + 2¹⁰) else if r > (p + 2⁹) then setting s = (r – 2¹⁰) else setting s = r;
- 5264 9. setting Originator Counter equal to ((q+s)*2³²)

14.6.4.1.6 Verifying the Originator Counter

5266 The Device shall verify the Originator Counter against the UTRN Counter Cache according
5267 to Section 14.3.6.

14.6.4.1.7 Deriving the Message Identifier

5269 The Device shall derive the Message Identifier by:

- 5270 • setting the Business Originator ID to the Entity Identifier in the Key Agreement Security
5271 Credentials it holds for the Trust Anchor Cell with Remote Party Role as supplier and
5272 cell usage of prePaymentTopUp;
- 5273 • setting the Business Target ID to its own Entity Identifier; and
- 5274 • setting Message Identifier to the concatenation Business Originator ID || Business
5275 Target ID || 0x01 || Originator Counter.

14.6.4.1.8 Validating the PTUT Supplier MAC

5277 The Device shall validate the PTUT Supplier MAC according to Section 14.3.5.

14.6.5 Response Construction

5279 The Device shall first update the UTRN Counter Cache according to Section 14.3.7.

³⁶ This derivation places a practical limit on the maximum increment between issued sequentially UTRN Counters. An increment of greater than (2⁹ - 1) between a UTRN Counter and the next one issued will cause this derivation to be inaccurate

- 5280 Where the Device is an ESME, the Device shall construct, and send via its HAN interface, a
 5281 Response message complying with the requirements of Use Case CS01a, using a Message
 5282 Identifier as specified in Section 14.6.4.1.7, and where the Originator Counter is as derived
 5283 by the calculations in Section 14.6.4.1.5.
- 5284 Where the Device is a GSME, the Device shall construct, and send via its HAN interface, a
 5285 Response message complying with the requirements of Use Case CS01b, using Message
 5286 Identifier as specified in Section 4.3.1.3, Message Identifier and where the Originator Counter
 5287 is as derived by the calculations in Section 14.6.4.1.5.

5288 **14.7 Applying a Prepayment Top Up to an ESME or GSME 5289 with consumer entry of a numeric code on a PPMID**

5290 **14.7.1 Description**

5291 This section covers the application of a Prepayment Top Up that has been bought for a
 5292 specific GSME or ESME to that GSME or ESME in the case where the consumer enters the
 5293 corresponding UTRN on a PPMID on the same SMHAN.

5294 The Use Case covering the Command is referenced in Section 14.7.4.1.2, The Use Case
 5295 covering the Response is referenced in Section 14.7.4.1.4.

5296 **14.7.2 Use Case Cross References**

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Command and Responses
Message Type Category	The Command and Response requirements are specifically as detailed in this Section 14.7
Capable of future dated invocation?	No
Protection Against Replay Required?	See Table 14.4.2
SEC User Gateway Services Schedule (Service Request) Reference	N/A
Valid Target Device(s)	ESME or GSME
Valid Business Originator role(s) for Command invocation (and so, for DLMS COSEM Commands, which Application Association is to be used for delivery of the Command to the Device) [Remote Party Messages Only]	Supplier -
Valid Response Recipient role(s) (only for Messages Authorised by the Access Control Broker on behalf of parties not known to the Device) [Remote Party Messages Only]	N/A
Valid initiating Device type(s) [SMHAN Only Messages]	N/A
Protocol	See this Section 14.7

5297 Table 14.7.2: Use Case Cross References for Prepayment Top Up through PPMID entry

5298 **14.7.3 Pre-conditions**

5299 None.

14.7.4 Detailed Steps

14.7.4.1 Detailed Steps / Sequence

14.7.4.1.1 Verifying the UTRN check digit

The PPMID may validate the 20th digit (the UTRN Check Digit) as specified at Section 14.8 (Calculating and Verifying the UTRN Check Digit). Where this check fails, the PPMID shall cease processing the Command and shall inform the consumer of the failure of the check digit.

14.7.4.1.2 Command Construction by the PPMID

Where the target Device is a GSME, the PPMID shall construct the Command according to the requirements of Use Case PCS01.

Where the target Device is an ESME, the PPMID shall construct a ZSE Consumer Top Up command.

In all cases:

- the value of the TopUp Code, with its ZSE meaning, shall be set to be a VisibleString whose value is the 20 digit UTRN; and
- the value of the Originating Device, with its ZSE meaning, shall be 0x02 (IHD).

14.7.4.1.3 HAN Only Command Validation by the ESME / GSME

If the ESME / GSME has no PPMID in its Device Log, the ESME / GSME shall apply the requirements of Section 6.2.4.2 and undertake no additional processing.

If the ESME / GSME has a PPMID in its Device Log:

- if the receiving Device is an ESME, the ESME shall use ZSE cryptographic processes to establish whether the Command was authentically issued by the PPMID that is in its Device Log; or
- if the receiving Device is a GSME, the GSME shall undertake Command Authenticity and Integrity Verification, as required for a Command of Message Category SME.C.PPMID-GSME to establish whether the Command was authentically issued by the PPMID that is in its Device Log.

If the Command was authentically issued by the PPMID within the Device Log, the ESME / GSME shall apply the requirements of Section 6.2.4.2.

If the Command was authentically issued by the PPMID within the Device Log, the ESME / GSME shall comply with the requirements of Section 14.6.4 (but excluding requirements in Sections 14.6.4.1.1, save that the ESME / GSME shall disregard the 20th digit before undertaking any further steps), and so process the contents of the Command accordingly.

14.7.4.1.4 HAN Only Response Construction and Issue

Where the ESME / GSME successfully creates a Remote Party Response to its Supplier, as per the requirements in Section 14.6.5, the ESME / GSME shall also:

- where the Device is a GSME, construct the HAN Only Response according to the requirements of Use Case PCS01 and send it to the PPMID; or
- where the Device is an ESME, construct a ZSE Consumer Top Up Response command, and send it to the PPMID.

In all cases the value of the Source of Top up, with its ZSE meaning, shall be 0x02 (IHD).

14.8 Calculating and Verifying the UTRN Check Digit

The UTRN Check Digit shall be calculated from the 19 decimal digit representation of the PTUT by a process equivalent to the following (Verhoeff's) Algorithm³⁷:

- setting an interim digit (referred to as IntDig) to have a value of zero;
- setting an index (referred to as K) to have a value of four;
- repeating the following steps with another index (referred to as J) taking the nineteen values of the integers from 1 to 19 in succession;
 - setting CurDig to the value of the J^{th} digit of the 19 decimal digits of the PTUT, where the first digit is the most significant (leftmost as written) and the nineteenth digit the least significant;
 - setting a third index (referred to as L) to the value in Table 14.8a using K as the Row Index and CurDig as the Column Index;
 - if the value of K is less than 7, setting K to the value of K+1, otherwise setting K to zero;
- q) setting IntDig to the value in Table 14.8b using IntDig as the Row Index and L as the Column Index;
- setting IntDig to the value in row 1 of Table 14.8c using the value of IntDig as the Column Index; and
- setting the UTRN Check Digit to the value of IntDig.

The UTRN Check Digit may be verified by undertaking exactly the same calculation on the 19 most significant digits of the UTRN, and comparing the result (the final value of *IntDig*, which would be used to set the UTRN Check Digit) to the 20th decimal digit which is the UTRN Check Digit.

		Column Index									
		0	1	2	3	4	5	6	7	8	9
Row Index	0	0	1	2	3	4	5	6	7	8	9
	1	1	5	7	6	2	8	3	0	9	4
	2	5	8	0	3	7	9	6	1	4	2
	3	8	9	1	6	0	4	3	5	2	7
	4	9	4	5	3	1	2	6	8	7	0
	5	4	2	8	6	5	7	3	9	0	1
	6	2	7	9	3	8	0	6	4	1	5
	7	7	0	4	6	9	1	3	2	5	8

Table 14.8a: Setting a third index

		Column Index									
		0	1	2	3	4	5	6	7	8	9
Row	0	0	1	2	3	4	5	6	7	8	9
	1	1	5	7	6	2	8	3	0	9	4

³⁷ See: (1) Verhoeff, J. (1969). *Error Detecting Decimal Codes (Tract 29)*. The Mathematical Centre, Amsterdam.
[doi:10.1002/zamm.19710510323](https://doi.org/10.1002/zamm.19710510323). (2) Kirtland, Joseph (2001). *Identification Numbers and Check Digit Schemes*. Mathematical Association of America. p. 153. [ISBN 0-88385-720-0](https://www.maa.org/maa-reviews/identification-numbers-and-check-digit-schemes). Retrieved August 26, 2011. (3) Salomon, David (2005). *Coding for Data and Computer Communications*. Springer. p. 56. [ISBN 0-387-21245-0](https://www.springer.com/978-0-387-21245-0). Retrieved August 26, 2011

Index	1	1	2	3	4	0	6	7	8	9	5
2	2	3	4	0	1	7	8	9	5	6	
3	3	4	0	1	2	8	9	5	6	7	
4	4	0	1	2	3	9	5	6	7	8	
5	5	9	8	7	6	0	4	3	2	1	
6	6	5	9	8	7	1	0	4	3	2	
7	7	6	5	9	8	2	1	0	4	3	
8	8	7	6	5	9	3	2	1	0	4	
9	9	8	7	6	5	4	3	2	1	0	

5366
5367

Table 14.8b: Setting IntDig using IntDig as a Row Index

		Column Index									
Row		0	1	2	3	4	5	6	7	8	9
Index	1	1	2	6	7	5	8	3	0	9	4

5368

Table 14.8c: Setting IntDig using IntDig as a Column Index

5369 15 Message Codes

- 5370 Message Codes shall be 2 octets in length and shall take the values specified in the 'Use
5371 Case reference' tab in the Mapping Table.
- 5372 For Messages specified by this GBCS, the most significant bit of the Message Code shall be
5373 0b0.

16 Event / Alert Codes and related requirements

16.1 Introduction – informative

This Section 16 sets out how Events and Alerts are handled. SMETS and CHTS define when Events occur and whether these Events are logged (in an Event Log) and additionally sent as an Alert via the HAN / WAN.

Table 16.2 defines Event Codes for events defined in SMETS and CHTS. It also indicates whether, as per SMETS and CHTS, there is a corresponding Alert issued over the Device's network interface (containing the relevant Event Code). It is important to note that not all Event Codes have a corresponding Alert. Where Alert Code is used elsewhere in this document, it equates to Event Code in Table 16.2.

Alerts sent over the SMHAN are not subject to the same message categorisation as those sent over the WAN. An Alert sent over the SMHAN is a native ZSE message.

16.1.1 Types of Alert

There are two Alert types. All have the same Grouping Header but different payloads as set out below:

- Alert type 1 - Payload comprises Alert Code and Timestamp only (two sub-types: DLMS and ZigBee). These are labelled 'Y(1)' in the 'Alert WAN (Alert type)' column in Table 16.2; and
- Alert type 2 - Payload comprises Alert Code, Timestamp and Use Case specific data as defined in Table 16.2 or main body of document (three sub-types: ASN.1, DLMS and ZigBee). These are labelled 'Y(2)' in the 'Alert WAN (Alert type)' column in Table 16.2.

Table 16.2 sets out the Alert type for each Alert Code. Examples of Use Case specific data include Billing Data Logs and content relating to future dated Commands (e.g. Message ID).

Table 16.2 sets out whether Alerts are mandated, mandatory conditional or non-mandated:

- Mandated - Alerts that Devices must support;
- Mandated conditional – Devices must support at least one from the specified group (e.g. there are seven Alerts in 'mandated – conditional group 1', Devices must support at least one of these seven); and
- Non-mandatory – no requirement for Devices to support, but where implemented Alert Codes shall have the meaning shown in Table 16.2. Further definition of these events may be found in the SSWG specifications³⁸.

16.1.2 Alert Construction

Alert construction is described in the GBCS in a number of places, including:

- Section 7.2.3 details common Message construction for all Alert types;
- Section 7.2.9 details Message construction for Alerts with DLMS COSEM Payloads. Table 7.2.9c details the required components of the Alert;
- Section 7.2.10 details Message construction for Alerts with ZSE Payloads. Table 7.2.10c details the required components of the Alert;

³⁸ Available from <http://www.triple-3.co.uk/sswg/>.

- 5413 • Sections 11.2 and 13.3 detail the Message Construction for the Alerts with ASN.1
5414 Payload; and
5415 • Section 9.2.2 details the Message Construction for future dated Alerts.

5416 **16.1.3 Event Behaviour**

5417 Detail on Event behaviour can be found in SMETS and CHTS using the relevant SMETS
5418 and CHTS reference in Table 16.2.

5419 **16.2 Event and Alert Codes**

5420 Table 16.2 lists the valid Event and Alert Codes, and sets out their requirements.



5421 GBCS v0.8.1 Event
5422 and Alert Codes.xlsx

5422 Table 16.2: Event and Alert Codes

5423 **16.3 Event Logs**

5424 Only GSME, ESME, CHF and GPF have Event Logs. The requirement set out in Table 16.2
5425 to log entries into Event Logs only applies to GSME, ESME, CHF and GPF as follows:

- 5426 • Event Log (GSME, ESME, CHF and GPF);
5427 • Security Event Log (GSME, ESME, CHF and GPF);
5428 • Power Event Log (ESME); and
5429 • ALCS Event Log (ESME).

5430 Use Cases to read logs (all) and clear logs (event logs only) are detailed in the Mapping
5431 Table.

5432 **16.4 Requirements**

5433 Event / Alert codes shall be 2 octets in length and shall take the values specified in Table
5434 16.2. As per the Device Specifications, all Alerts, Event Log entries, Security Log entries,
5435 Power Event Log entries and ALCS Event Log entries shall contain a UTC date time stamp,
5436 in addition to the Event / Alert code. Non-Critical Alerts can be configured to be sent / not to
5437 be sent using the relevant Commands and Responses defined in Use Cases ECS25a,
5438 ECS25b and GCS20 (all configurable Alerts can be configured in a single Message). The
5439 relevant DCC User needs to ensure that Critical Alerts are always configured on.

5440 As specified in Table 16.2 by way of 'x' in a cell, deviceType (and for ESME, variant of
5441 ESME) shall determine which Alerts a device shall issue and which Event Log and Security
5442 Log entries it shall record. Where deviceType = 0x04 (HCALCS) or 0x05 (PPMID), this
5443 Section 16 only requires the sending of Alerts, since neither Device type is required to have
5444 either an Event Log or a Security Log.

5445 Where an Alert and a Log entry have the same trigger in a Device, the Device shall record
5446 the same UTC date time stamp and the same Event / Alert code in both.

5447 The Remote Party to which an Alert containing a specific Event Code is addressed shall be
5448 determined by the Remote Party Role as specified in Table 16.2. Where the Remote Party
5449 Role is stated as Supplier or WAN Provider, the Alert shall be addressed:

- 5450 • to the WAN Provider if deviceType = 0x02 (CHF); and
5451 • to the Supplier for all other deviceType values.

- 5452 Where a Use Case is specified in Table 16.2 the corresponding Alert shall be constructed
 5453 according to the specified Use Case. Where no Use Case is specified the Alert shall be
 5454 constructed according to Section 7.
- 5455 Where an Alert has two recipient roles identified, the Device shall place the Entity ID of the
 5456 Supplier in the Business Target ID field and the Entity ID of the other recipient in the
 5457 Supplementary Remote Party ID field.
- 5458 For any Event Log entries relating to Event Codes 0x0061 and 0x0062, the Device shall
 5459 record the commands input on the User Interface by including the User Interface Command
 5460 Code in the Event Log entry as defined in Table 16.4.

User Interface Command Code	User Interface Command (from SMETS)	GSME	ESME	ESME with ALCS	ESME with Boost Function
0x0001	Activate Boost Period				x
0x0002	Activate Emergency Credit [PIN]	x	x		
0x0005	Add Credit	x	x		
0x0008	Allow Access to User Interface	x	x		
0x0009	Arm Supply	x	x		
0x000A	Cancel Boost Period				x
0x000B	Check for HAN Interface Commands	x			
0x000C	Disable Privacy PIN Protection [PIN]	x	x		
0x000E	Enable Supply [PIN]	x	x		
0x000F	Extend Boost Period				x
0x0012	Set Privacy PIN [PIN]	x	x		
0x0013	Test Auxiliary Load Control Switch 1			x	
0x0014	Test Auxiliary Load Control Switch 2			x	
0x0015	Test Auxiliary Load Control Switch 3			x	
0x0016	Test Auxiliary Load Control Switch 4			x	
0x0017	Test Auxiliary Load Control Switch 5			x	
0x0018	Test Valve	x			
0x0019	Reset Remaining Battery Capacity	x			
0x001A	Find and Join SMHAN	x	x	x	x

5461 Table 16.4: User Interface Command Codes by Device For any Event Log entries relating to
 5462 Event Codes 0x0054 and 0x0055, the Device shall record the Commands received on the
 5463 Network Interface by including the Message Code in the Event Log.

17 Remote Party Usage Rights

17.1 Remote Party Access Rights to Attributes and Methods

Access rights to attributes and methods shall be enforced by the Device as per the requirements in the 'SMETS required objects' tab in the Mapping Table. 'R' shall mean that the Remote Party Role shall have read access to the attribute. 'W' shall mean that the Remote Party Role shall have write access to the attribute. 'A' shall mean that the Remote Party Role shall be able to invoke the method. There shall be no other access to these attributes and methods allowed by the Device.

Encryption of attributes whenever transiting the HAN Interface shall be enforced by the Device as per the requirements in the 'SMETS required objects' tab in the Mapping Table. 'Y' in the column headed 'Encrypted' shall mean that the Encryption shall always be applied to the corresponding attribute as it crosses the HAN Interface.

17.2 Remote Party Usage Rights to Use Cases

Access rights to Use Cases shall be enforced by the Device as per the requirements in the Use Case Access Permissions table in each Use Case (see Table 19.4). In that table, 'A' shall mean that the Remote Party Role shall have access to the Use Case. There shall be no other access allowed by the Device. Remote Party roles align to the Trust Anchor Cells in Section 4.3.2.5. The Access Control Broker controls access for Unknown Remote Parties.

5483 18 Message Templates

5484 18.1 GBZ and ZSE Message Templates

5485 Message Templates for GBZ Use Cases are detailed in the embedded Use Cases, Section
 5486 19.3. These Message Templates are derived from the Mapping Table, and shall be
 5487 complied with in the construction and population of all such Messages.

5488 18.1.1 Message Templates for ZSE commands between ESME and 5489 HCALCS

5490 18.1.1.1 ZSE Load Control Event command

5491 The ZSE Load Control Event command shall be sent by an ESME, on:

- 5492 • successful authentication of a Command with Message Code 0x0055;
- 5493 • to control a HCALCS according to the Auxiliary Load Control Switch Calendar; or
- 5494 • on receipt of a Get Scheduled Events command from an HCALCS where required by
 5495 SMETS.

5496 In executing this command, the ESME shall send the ZCL Load Control Event command to
 5497 the HCALCS identified in that Command with:

- 5498 • the values of each field populated in the ZCL Load Control Event command as specified
 5499 in Table 18.1.1.1;
- 5500 • the ‘Duration in Minutes’ field set according to the respective triggers above:
 - 5501 ○ the duration specified in the Command with Message Code 0x0056;
 - 5502 ○ the duration of the command defined in the Auxiliary Load Control Switch Calendar;
 5503 or
 - 5504 ○ the remaining duration calculated as per SMETS.
- 5505 • the ‘Duty Cycle’ field set to 0x00, where the Command specifies that the switch is to be
 5506 turned off; and
- 5507 • the ‘Duty Cycle’ field set to 0x64, where the Command specifies that the switch is to be
 5508 turned on.

5509 The recipient HCALCS shall interpret the value in Duty Cycle accordingly.

5510 On successful authentication of such a ZCL command, the recipient HCALCS shall respond
 5511 with a Report Event Status ZCL command populated as per Table 18.1.1.4, with Event
 5512 Status set to:

- 5513 • 0x02 (‘Event started’), if the command was successfully executed; or
- 5514 • 0xFE (‘Load Control Event command Rejected’), if the command was not successfully
 5515 executed.

Element	Meaning	Value	Octets
ZCL header			
Frame control	Cluster-specific; not manufacturer specific; server-client; allow default response;	0b00001001	1
Transaction sequence		0x00	1

number			
Command identifier	Load Control Event	0x00	1
ZCL payload			
Issuer Event ID (UINT32)	Set to the ESME's current UTC time	See 'Meaning' column	4
Device Class (BITMAP16)	All device types	0xFFFF	2
Utility Enrollment Group (UINT8)	All groups	0x00	1
Start Time (UTCTime)	Start immediately	0x00000000	4
Duration In Minutes (UINT16)	A value between 1 and 1440 minutes	See 'Meaning' column	2
Criticality Level (UINT8)	Voluntary	0x01	1
Cooling Temperature Offset (UINT8)	Not used	0xFF	1
Heating Temperature Offset (UINT8)	Not used	0xFF	1
Cooling Temperature Set Point (INT16)	Not used	0x8000	2
Heating Temperature Set Point (INT16)	Not used	0x8000	2
Average Load Adjustment Percentage (INT8)	Not used	0x80	1
Duty Cycle (UINT8)	0x00 (0) = switch OFF; 0x64 (100) = switch ON	See 'Meaning' column	1
Event Control (BITMAP8)	Do not randomise	0x00	1

5516 Table 18.1.1.1: ZSE Load Control Event command

5517 **18.1.1.2 ZSE Cancel Load Control Event command**

5518 The ZCL Cancel Load Control Event command shall be sent by an ESME, on successful
 5519 authentication of a Command with Message Code 0x0057, to the Device identified in that
 5520 Command with the values of each field populated in the ZCL Cancel Load Control Event
 5521 command as specified in Table 18.1.1.2.

5522 On successful authentication of such a ZCL Command, the recipient HCALCS shall respond
 5523 with a Report Event Status ZCL command populated as per Table 18.1.1.4, with Event
 5524 Status set to:

- 5525 • 0x06 ('The event has been cancelled'), if the command successfully cancelled the
 5526 specified Load Control Event; or
- 5527 • 0xFD ('Rejected - Invalid Cancel Command (Undefined Event)'), if the Load Control
 5528 Event had already ended; or

- 5529 • 0xF8 ('Rejected - Invalid Cancel Command (Default)'), if the command failed to execute.

Element	Meaning	Value	Octets
ZCL header			
Frame control	Cluster-specific; not manufacturer specific; server-client; disable default response;	0b00011001	1
Transaction sequence number		0x00	1
Command identifier	Cancel Load Control Event	0x01	1
ZCL payload			
Issuer Event ID (UINT32)	Set to the value of the Issuer Event ID in the corresponding ZSE Load Control Event command	See 'Meaning' column	4
Device Class (BITMAP16)	All device types	0xFFFF	2
Utility Enrollment Group (UINT8)	All groups	0x00	1
Cancel Control (BITMAP8)	No randomisation by the device	0x00	1
Effective Time (UTCTime)	Fixed by the ZSE specification	0x00000000	4

5530 Table 18.1.1.2: ZSE Cancel Load Control Event command

5531 **18.1.1.3 ZSE Get Scheduled Event command**

5532 When sending a ZSE Get Scheduled Event command pursuant to Section 8.5.2.1 of SMETS,
 5533 an HCALCS shall populate that ZCL Command according to Table 18.1.1.3.

5534 On authenticated receipt of ZSE Get Scheduled Event command, the ESME shall send a
 5535 ZSE Load Control Event command instructing the HCALCS whether it is to be open or
 5536 closed, and for how long it is to be in that state.

Element	Meaning	Value	Octets
ZCL header			
Frame control	Cluster-specific; not manufacturer specific; client-server; disable default response;	0b00010001	1
Transaction sequence number		0x00	1
Command identifier	Get Scheduled Events	0x01	1
ZCL payload			
Start Time (UTCTime)	Retrieve active event	0x00000000	4

Number of Events (UINT8)	Device can only accept 1 event	0x01	1
--------------------------	--------------------------------	------	---

Table 18.1.1.3: ZSE Get Scheduled Event command

5537

18.1.1.4 ZSE Report Event Status command

Element	Meaning	Value	Octets
ZCL header			
Frame control	Cluster-specific; not manufacturer specific; client-server; allow default response;	0b00000001	1
Transaction sequence number		0x00	1
Command identifier	Report Event Status	0x00	1
ZCL payload			
Issuer Event ID (UINT32)	Set to the event ID from the corresponding ZSE command received from the ESME	See 'Meaning' column	4
Event Status (UINT8)	Refer to ZigBee standard	As per the requirements of this Section 18.1.1	1
Event Status Time (UTCTime)	An HCALCS is not required to have a clock and therefore the HCALC is not required to know UTC time	0x00000001	4
Criticality Level Applied (UINT8)	0x01 = Voluntary	0x01	1
Cooling Temperature Set Point Applied (UINT16)	Not used	0x8000	2
Heating Temperature Set Point Applied (UINT16)	Not used	0x8000	2
Average Load Adjustment Percentage Applied (INT8)	Not used	0x80	1
Duty Cycle Applied (UINT8)	0x00 (0) = switched OFF; 0x64 (100) = switched ON	See 'Meaning' column	1
Event Control (BITMAP8)	Do not randomise	0x00	1
Signature Type (UINT8)	No signature	0x00	1

Table 18.1.1.4: ZSE Report Event Status command

5538

18.2 DLMS COSEM Message Templates

5539

Table 18.2 contains Message Templates for all Use Case with DLMS COSEM payloads.

5540

These Message Templates are derived from the Mapping Table, and shall be complied with

5541

in the construction and population of all such Messages.



GBCS v0.8.1 DLMS
COSEM Message Tem

5542

5543 Table 18.2: DLMS COSEM Message Templates

5544 **18.2.1 Encoding**

5545 Italicised terms in this Section 18.2.1 shall have their DLMS COSEM meaning.

5546 **18.2.1.1 Compact array encoding**

5547 The Blue Book definition of attribute 2 of *Profile Generic* objects may be interpreted as
5548 requiring ‘entry’ to be a *structure* containing a single choice from the DLMS data types. The
5549 GBCS interprets it as meaning that ‘entry’ is a *structure* that can contain multiple choices of
5550 DLMS data types. These choices vary between instances of Profile Generic object. To
5551 identify these different structures, the naming convention ‘entry_nameOfStructure’ is used.

5552 The GBCS uses the *compact-array* data type in attribute 2 of *Profile Generic* objects. Table
5553 18.2.1.1 details the derivation of the *contents-description* element within the *compact-array*
5554 *structure* for the structures used in the *Profile Generic* objects required by this GBCS. These
5555 encodings are reflected in the DLMS COSEM Message Templates.

Structure definition	Tag	Number of entries (structures and arrays only)	Tag of entries in array	contents-description for compact-array
entry_dlValueLogEntry ::= structure { timestamp: double-long-unsigned, dlValue: double-long-unsigned }	0x02 0x06 0x06	0x02		0x1302020606
entry_enumValueLogEntry ::= structure { timestamp: double-long-unsigned, enumValue: enum }	0x02 0x06 0x16	0x02		0x1302020616
entry_eventLogEntry12 ::= structure { timestamp: double-long-unsigned, logCode: long-unsigned, otherInformation: octet-string(12) }	0x02 0x06 0x12 0x09	0x03		0x130203061209
entry_powerLogEntry ::= structure { timestamp: double-long-unsigned, logCode: long-unsigned, otherInformation: double-long-unsigned }	0x02 0x06 0x12 0x06	0x03		0x130203061206
entry_eventLogEntry8 ::= structure { timestamp: double-long-unsigned,	0x02 0x06	0x03		0x130203061209

Structure definition	Tag	Number of entries (structures and arrays only)	Tag of entries in array	contents-description for compact-array
logCode: long-unsigned,	0x12			
otherInformation: octet-string(8)	0x09			
}				
entry_securityLogEntry ::= structure {	0x02	0x02		0x1302020612
timestamp: double-long-unsigned,	0x06			
logCode: long-unsigned	0x12			
}				
entry_billingCalendarLogEntry ::= structure{	0x02	0x07 or 0x09		0x1302070606013006010806010806010806010806 (single element) or 0x130209060606013006010406010806010806010806010806 (twin element)
timestamp: double-long-unsigned,	0x06			
activeImportRegisterValue: double-long-unsigned,	0x06			
secondaryActiveImportRegisterValue: double-long-unsigned, [[MAY NOT BE PRESENT]]	0x06			
tariffTOURegisterValues: array double-long-unsigned,	0x01	0x30	0x06	
secondaryTariffTOURegisterValues: array double-long-unsigned, [[MAY NOT BE PRESENT]]	0x01	0x04	0x06	
tariffTOUBlock1RegisterValues: array double-long-unsigned,	0x01	0x08	0x06	
tariffTOUBlock2RegisterValues: array double-long-unsigned,	0x01	0x08	0x06	

Structure definition	Tag	Number of entries (structures and arrays only)	Tag of entries in array	contents-description for compact-array
tariffTOUBlock3RegisterValues: array double-long-unsigned,	0x01	0x08	0x06	
tariffTOUBlock4RegisterValues: array double-long-unsigned	0x01	0x08	0x06	
}				
entry_billingCalendarOnSetModeOrTariffLogEntry::= structure{	0x02	0x0D or 0x0F		0x13020D06060130060108060108060108060108060108060505050505 (single element) or 0x13020F060606013006010406010806010806010806050505050505 (twin element)
timestamp: double-long-unsigned,	0x06			
activeImportRegisterValue: double-long-unsigned,	0x06			
secondaryActiveImportRegisterValue: double-long-unsigned, [[MAY NOT BE PRESENT]]	0x06			
tariffTOURegisterValues: array double-long-unsigned,	0x01	0x30	0x06	
secondaryTariffTOURegisterValues: array double-long-unsigned, [[MAY NOT BE PRESENT]]	0x01	0x04	0x06	
tariffTOUBlock1RegisterValues: array double-long-unsigned,	0x01	0x08	0x06	
tariffTOUBlock2RegisterValues: array double-long-unsigned,	0x01	0x08	0x06	
tariffTOUBlock3RegisterValues: array double-long-unsigned,	0x01	0x08	0x06	
tariffTOUBlock4RegisterValues: array double-long-unsigned	0x01	0x08	0x06	
emergencyCreditBalanceValue: double-long,	0x05			
meterBalanceValue: double-long,	0x05			

Structure definition	Tag	Number of entries (structures and arrays only)	Tag of entries in array	contents-description for compact-array
paymentDebtRegisterValue: double-long,	0x05			
timeDebtRegisters1Value: double-long,	0x05			
timeDebtRegisters2Value: double-long,	0x05			
accumulatedDebtRegisterValue: double-long	0x05			
}				
entry_boostFunctionLogEntry ::= structure {	0x02	0x02		0x1302020606
boost_start: double-long-unsigned,	0x06			
boost_end: double-long-unsigned	0x06			
}				
entry_prepaymentReadLogEntry ::= structure {	0x02	0x07		0x13020706050505050505
timestamp: double-long-unsigned,	0x06			
emergencyCreditBalanceValue: double-long,	0x05			
meterBalanceValue: double-long,	0x05			
paymentDebtRegisterValue: double-long,	0x05			
timeDebtRegisters1Value: double-long,	0x05			
timeDebtRegisters2Value: double-long,	0x05			
accumulatedDebtRegisterValue: double-long	0x05			
}				
entry_registerReadLogEntry ::= structure{	0x02	0x07 or 0x09		0x1302070606013006010806010806010806010806 (single element) or

Structure definition	Tag	Number of entries (structures and arrays only)	Tag of entries in array	contents-description for compact-array
				0x130209060606013006010406010806010806010806010806 (twin element)
timestamp: double-long-unsigned,	0x06			
activeImportRegisterValue: double-long-unsigned,	0x06			
secondaryActiveImportRegisterValue: double-long-unsigned, [[MAY NOT BE PRESENT]]	0x06			
tariffTOURRegisterValues: array double-long-unsigned,	0x01	0x30	0x06	
secondaryTariffTOURRegisterValues: array double-long-unsigned, [[MAY NOT BE PRESENT]]	0x01	0x04	0x06	
tariffTOUBlock1RegisterValues: array double-long-unsigned,	0x01	0x08	0x06	
tariffTOUBlock2RegisterValues: array double-long-unsigned,	0x01	0x08	0x06	
tariffTOUBlock3RegisterValues: array double-long-unsigned,	0x01	0x08	0x06	
tariffTOUBlock4RegisterValues: array double-long-unsigned	0x01	0x08	0x06	
}				
entry_activeImportLogEntry ::= structure {	0x02	0x03 or 0x02		0x130203060606 (twin element) or 0x1302020606 (single element)
timestamp: double-long-unsigned,	0x06			
primaryValue: double-long-unsigned,	0x06			
secondaryValue: double-long-unsigned [[MAY NOT BE PRESENT]]	0x06			
}				
entry_twoDIValueLogEntry ::= structure {	0x02	0x03		0x130203060606

Structure definition	Tag	Number of entries (structures and arrays only)	Tag of entries in array	contents-description for compact-array
timestamp: double-long-unsigned,	0x06			
dValue: double-long-unsigned,	0x06			
dValue2: double-long-unsigned	0x06			
}				
entry_alcsLogEntry::= structure {	0x02	0x04		0x13020406061606
timestamp: double-long-unsigned,	0x06			
switchNumberAndAction: double-long-unsigned,	0x06			
outcome: enum,	0x16			
hANCommandID: double-long-unsigned	0x06			
}				

5556 Table18.2.1.1: derivation of the *contents-description* element within the *compact-array structure*

5557 **18.2.1.2 Values of the *credit_charge_configuration* attribute of Account (Class ID 111) objects**

5558 There are three SMETS parameters required for all Set Payment Mode Use Cases:

- 5559 • Payment Mode, being Credit or Prepayment;
- 5560 • Suspend Debt Emergency, being True or False and only being relevant when Payment Mode = Prepayment; and
- 5561 • Suspend Debt Disabled, being True or False and only being relevant when Payment Mode = Prepayment.

5562 Note that Disablement Threshold can also be set through the 'Set Payment Mode to Prepayment' Use Case.

5563 The combination of these values determines, and is reflected in, the five possible values in the *credit_charge_configuration* attribute of the
5564 Account objects.

5565 On an ESME that is not a twin element variant, the ESME shall accept only the five values for the *credit_charge_configuration* attribute set out
 5566 in Table 18.2.1.2a.

Payment Mode	Suspend Debt Emergency	Suspend Debt Disabled	Value of credit_charge_configuration attribute	Length in octets
Credit	Not relevant	Not relevant	0x0102 020309060000130A00FF09060000130200FF0403E0 020309060000130A00FF09060000130204FF0403E0	46
Prepayment	True	True	0x010D 020309060000130A00FF09060000130200FF0403E0 020309060000130A00FF09060000130204FF0403E0 020309060000130A00FF09060000130201FF0403C0 020309060000130A00FF09060000130202FF0403C0 020309060000130A00FF09060000130203FF0403E0 020309060000130A01FF09060000130203FF0403E0 020309060000130A01FF09060000130200FF0403E0 020309060000130A01FF09060000130204FF0403E0 020309060000130A01FF09060000130201FF0403C0 020309060000130A01FF09060000130202FF0403C0 020309060000130A02FF09060000130204FF0403E0 020309060000130A02FF09060000130201FF0403E0 020309060000130A02FF09060000130202FF0403E0	275
Prepayment	True	False	0x010D 020309060000130A00FF09060000130200FF0403E0 020309060000130A00FF09060000130204FF0403E0 020309060000130A00FF09060000130201FF0403E0 020309060000130A00FF09060000130202FF0403E0 020309060000130A00FF09060000130203FF0403E0 020309060000130A01FF09060000130203FF0403E0 020309060000130A01FF09060000130200FF0403E0 020309060000130A01FF09060000130204FF0403E0 020309060000130A01FF09060000130201FF0403E0 020309060000130A01FF09060000130202FF0403E0 020309060000130A02FF09060000130204FF0403E0 020309060000130A02FF09060000130201FF0403E0 020309060000130A02FF09060000130202FF0403E0	275
Prepayment	False	True	0x010A	212

Payment Mode	Suspend Debt Emergency	Suspend Debt Disabled	Value of credit_charge_configuration attribute	Length in octets
			020309060000130A00FF09060000130200FF0403E0 020309060000130A00FF09060000130204FF0403E0 020309060000130A00FF09060000130201FF0403C0 020309060000130A00FF09060000130202FF0403C0 020309060000130A00FF09060000130203FF0403E0 020309060000130A01FF09060000130203FF0403E0 020309060000130A01FF09060000130200FF0403E0 020309060000130A01FF09060000130204FF0403E0 020309060000130A01FF09060000130201FF0403C0 020309060000130A01FF09060000130202FF0403C0	
Prepayment	False	False	0x010A 020309060000130A00FF09060000130200FF0403E0 020309060000130A00FF09060000130204FF0403E0 020309060000130A00FF09060000130201FF0403E0 020309060000130A00FF09060000130202FF0403E0 020309060000130A00FF09060000130203FF0403E0 020309060000130A01FF09060000130203FF0403E0 020309060000130A01FF09060000130200FF0403E0 020309060000130A01FF09060000130204FF0403E0 020309060000130A01FF09060000130201FF0403E0 020309060000130A01FF09060000130202FF0403E0	212

5567 Table 18.2.1.2a: allowable values for the *credit_charge_configuration* attribute for all ESME (except twin element variant)

5568 On an ESME that is a twin element variant, the ESME shall accept only the five values for the *credit_charge_configuration* attribute in Table
5569 18.2.1.2b.

Payment Mode	Suspend Debt Emergency	Suspend Debt Disabled	Value of credit_charge_configuration attribute	Length in octets
Credit	Not relevant	Not relevant	0x0103 020309060000130A00FF09060000130200FF0403E0 020309060000130A00FF09060000130204FF0403E0 020309060000130A00FF09060000130205FF0403E0	65
Prepayment	True	True	0x010F 020309060000130A00FF09060000130200FF0403E0	317

Payment Mode	Suspend Debt Emergency	Suspend Debt Disabled	Value of credit_charge_configuration attribute	Length in octets
			020309060000130A00FF09060000130204FF0403E0 020309060000130A00FF09060000130201FF0403C0 020309060000130A00FF09060000130202FF0403C0 020309060000130A00FF09060000130203FF0403E0 020309060000130A01FF09060000130203FF0403E0 020309060000130A01FF09060000130200FF0403E0 020309060000130A01FF09060000130204FF0403E0 020309060000130A01FF09060000130201FF0403C0 020309060000130A01FF09060000130202FF0403C0 020309060000130A02FF09060000130204FF0403E0 020309060000130A02FF09060000130201FF0403E0 020309060000130A02FF09060000130202FF0403E0 020309060000130A00FF09060000130205FF0403E0 020309060000130A01FF09060000130205FF0403E0	
Prepayment	True	False	0x010F 020309060000130A00FF09060000130200FF0403E0 020309060000130A00FF09060000130204FF0403E0 020309060000130A00FF09060000130201FF0403E0 020309060000130A00FF09060000130202FF0403E0 020309060000130A00FF09060000130203FF0403E0 020309060000130A01FF09060000130203FF0403E0 020309060000130A01FF09060000130200FF0403E0 020309060000130A01FF09060000130204FF0403E0 020309060000130A01FF09060000130201FF0403E0 020309060000130A01FF09060000130202FF0403E0 020309060000130A02FF09060000130204FF0403E0 020309060000130A02FF09060000130201FF0403E0 020309060000130A02FF09060000130202FF0403E0 020309060000130A00FF09060000130205FF0403E0 020309060000130A01FF09060000130205FF0403E0	317
Prepayment	False	True	0x010C 020309060000130A00FF09060000130200FF0403E0 020309060000130A00FF09060000130204FF0403E0 020309060000130A00FF09060000130201FF0403C0 020309060000130A00FF09060000130202FF0403C0	264

<i>Payment Mode</i>	<i>Suspend Debt Emergency</i>	<i>Suspend Debt Disabled</i>	<i>Value of credit_charge_configuration attribute</i>	<i>Length in octets</i>
			020309060000130A00FF09060000130203FF0403E0 020309060000130A01FF09060000130203FF0403E0 020309060000130A01FF09060000130200FF0403E0 020309060000130A01FF09060000130204FF0403E0 020309060000130A01FF09060000130201FF0403C0 020309060000130A01FF09060000130202FF0403C0 020309060000130A00FF09060000130205FF0403E0 020309060000130A01FF09060000130205FF0403E0	
Prepayment	False	False	0x010C 020309060000130A00FF09060000130200FF0403E0 020309060000130A00FF09060000130204FF0403E0 020309060000130A00FF09060000130201FF0403E0 020309060000130A00FF09060000130202FF0403E0 020309060000130A00FF09060000130203FF0403E0 020309060000130A01FF09060000130203FF0403E0 020309060000130A01FF09060000130200FF0403E0 020309060000130A01FF09060000130204FF0403E0 020309060000130A01FF09060000130201FF0403E0 020309060000130A01FF09060000130202FF0403E0 020309060000130A00FF09060000130205FF0403E0 020309060000130A01FF09060000130205FF0403E0	264

Table 18.2.1.2b: allowable values for the *credit_charge_configuration* attribute for all ESME (twin element variant)5570 **18.2.1.3 Deriving the values of the credit_charge_configuration attribute of Account (Class ID 111) objects - informative**

5571 This section explains the derivation of the five values of this attribute that an ESME can accept.

5572 The *credit_charge_configuration* attribute encoding is shown in Table 18.2.1.3a.

<i>Component</i>	<i>Hex value</i>	<i>Length in octets</i>	<i>Notes</i>
credit_charge_configuration			
Tag	0x01	1	tag for array
Length	Variable	1	entries in array

Component	Hex value	Length in octets	Notes
credit_charge_configuration_element			
Tag	0x02	1	
Length	0x03	1	3 elements in this structure
credit_reference			
Tag	0x09	1	tag for octet-string
Length	0x06	1	logical_name is 6 octets
Value	Variable	6	OBIS code for this class 112 object
charge_reference			
Tag	0x09	1	tag for octet-string
Length	0x06	1	logical_name is 6 octets
Value	Variable	6	OBIS code for this class 113 object
collection_configuration			
Tag	0x04	1	tag for bit-string
Length	0x03	1	3 as per the Blue Book
Value	0b11Z	1	Where Z is the variable Bit 0;
trailing_bits	0b00000	1	

5573 Table 18.2.1.3a: *credit_charge_configuration* attribute encoding So the value of the *credit_charge_configuration_element* attribute is a 21 octet long
 5574 concatenation:

5575 0x02030906 || credit object OBIS code || 0x0906 || charge object OBIS code || 0x0403 || collection bit string || 0b00000

5576 The meaning of each *credit_charge_configuration_element* is that this charge can be collected from this credit object, except in possible meter
 5577 states specified by the *collection_configuration* bit string.

5578 On an ESME, there shall be three class 112 Credit objects, as shown in Table 18.2.1.3b. Two are not relevant in Credit Mode.

SMET Reference Component	OBIS Code (decimal)	OBIS Code (hexadecimal)	Payment Mode
MeterBalance	0-0:19.10.0.255	0x0000130A00FF	Prepayment and Credit

SMET Reference Component	OBIS Code (decimal)	OBIS Code (hexadecimal)	Payment Mode
AccumulatedDebt	0-0:19.10.2.255	0x0000130A02FF	Prepayment
EmergencyCreditBalance	0-0:19.10.1.255	0x0000130A01FF	Prepayment

5579 Table 18.2.1.3b: Class 112 Credit objects

5580 There shall be five class 113 Charge objects on an ESME (or six on a twin element ESME), as shown in Table 18.2.1.3c. Three are not
 5581 relevant in Credit Mode.

SMET Reference Component	OBIS Code (decimal)	OBIS Code (hexadecimal)	Payment Mode
DebtRecoveryRates[1]	0-0:19.2.1.255	0x0000130201FF	Prepayment
DebtRecoveryRates[2]	0-0:19.2.2.255	0x0000130202FF	Prepayment
DebtRecoveryPerPayment	0-0:19.2.3.255	0x0000130203FF	Prepayment
SecondaryTariffTOUPriceMatrix (Twin element ESME only)	0-0:19.2.5.255	0x0000130205FF	Prepayment and Credit
StandingCharge	0-0:19.2.4.255	0x0000130204FF	Prepayment and Credit
TariffBlockPriceMatrixTOU	0-0:19.2.0.255	0x0000130200FF	Prepayment and Credit

5582 Table 18.2.1.3c: Class 113 Charge objects As defined in the Blue Book, the *collection_configuration* bit string determines whether a charge is
 5583 collected from a credit dependent on ESME state.

5584 Bit 1 affects charging in load limiting periods. There is no such requirement in SMETS, so this value is always 0b1 (charges are applied in load
 5585 limiting periods).

5586 In Credit Mode, collection continues in all states, so the value of all three bits is always 0b1.

5587 In Prepayment Mode, *collection_configuration* is set according to Suspend Debt Disabled (affects Bit 0) values, and the pairing of charge and
 5588 credit object.

5589 Suspend Debt Emergency being True means that DebtRecoveryRates[1..2] and StandingCharge are collected from AccumulatedDebt rather
 5590 than EmergencyCreditBalance, when Emergency Credit is in use, so Suspend Debt Emergency is specified by way of pairing charge and credit
 5591 objects accordingly. Note that Bit 2 of *collection_configuration* shall always be fixed at 0b1.

5592 Suspend Debt Disabled being True means that DebtRecoveryRates[1..2] are no longer collected when the supply is disabled due to lack of
 5593 credit.

5594 Table 18.2.1.3d sets out the *credit_charge_configuration_element* array entries in Credit Mode.

<i>Tag & Length</i>	<i>Credit object</i>	<i>Tag & Length</i>	<i>Charge object</i>	<i>Tag & Length</i>	<i>Collection bit string</i>	<i>trailing_bits</i>	<i>Array entry</i>
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130200FF (TariffBlockPriceMatrixTOU)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130200FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130204FF (StandingCharge)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130204FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130205FF (SecondaryTariffTOUPriceMatrix (Twin element ESME only))	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130205FF0403E0

5595 Table 18.2.1.3d: Class 113 Charge objects

5596 When the mode is set as in Table 18.2.1.3e:

<i>Payment Mode</i>	<i>Suspend Debt Emergency</i>	<i>Suspend Debt Disabled</i>
Prepayment	False	False

5597 Table 18.2.1.3e: Prepayment states

5598 the *credit_charge_configuration_element* array entries are as per Table 18.2.1.3f.

<i>Tag & Length</i>	<i>Credit object</i>	<i>Tag & Length</i>	<i>Charge object</i>	<i>Tag & Length</i>	<i>Collection bit string</i>	<i>trailing_bits</i>	<i>Array entry</i>
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130200FF (TariffBlockPriceMatrixTOU)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130200FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130204FF (StandingCharge)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130204FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130201FF (DebtRecoveryRates[1])	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130201FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130202FF (DebtRecoveryRates[2])	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130202FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130203FF (DebtRecoveryPerPayment)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130203FF0403E0

<i>Tag & Length</i>	<i>Credit object</i>	<i>Tag & Length</i>	<i>Charge object</i>	<i>Tag & Length</i>	<i>Collection bit string</i>	<i>trailing_bits</i>	<i>Array entry</i>
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130203FF (DebtRecoveryPerPayment)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130203FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130200FF (TariffBlockPriceMatrixTOU)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130200FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130204FF (StandingCharge)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130204FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130201FF (DebtRecoveryRates[1])	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130201FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130202FF (DebtRecoveryRates[2])	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130202FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130205FF (SecondaryTariffTOUPriceMatrix (Twin element ESME only))	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130205FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130205FF (SecondaryTariffTOUPriceMatrix (Twin element ESME only))	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130205FF0403E0

5599 Table 18.2.1.3f: *credit_charge_configuration_element* array entries Note that, as per SMETS, the value of MeterBalance determines whether
 5600 charges are collected from EmergencyCreditBalance or MeterBalance.

5601 When the mode is set as in Table 18.2.1.3g:

<i>Payment Mode</i>	<i>Suspend Debt Emergency</i>	<i>Suspend Debt Disabled</i>
Prepayment	False	True

5602 Table 18.2.1.3g: Prepayment states

5603 the *credit_charge_configuration_element* array entries are as per Table 18.2.1.3h.

<i>Tag & Length</i>	<i>Credit object</i>	<i>Tag & Length</i>	<i>Charge object</i>	<i>Tag & Length</i>	<i>Collection bit string</i>	<i>trailing_bits</i>	<i>Array entry</i>
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130200FF (TariffBlockPriceMatrixTOU)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130200FF0403E0

<i>Tag & Length</i>	<i>Credit object</i>	<i>Tag & Length</i>	<i>Charge object</i>	<i>Tag & Length</i>	<i>Collection bit string</i>	<i>trailing_bits</i>	<i>Array entry</i>
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130204FF (StandingCharge)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130204FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130201FF (DebtRecoveryRates[1])	0x0403	0b110 (do not collect when supply is disabled due to no credit)	0b00000	0x020309060000130A00FF09060000130201FF0403C0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130202FF (DebtRecoveryRates[2])	0x0403	0b110 (do not collect when supply is disabled due to no credit)	0b00000	0x020309060000130A00FF09060000130202FF0403C0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130203FF (DebtRecoveryPerPayment)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130203FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130203FF (DebtRecoveryPerPayment)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130203FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130200FF (TariffBlockPriceMatrixTOU)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130200FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130204FF (StandingCharge)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130204FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130201FF (DebtRecoveryRates[1])	0x0403	0b110 (do not collect when supply is disabled due to no credit)	0b00000	0x020309060000130A01FF09060000130201FF0403C0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130202FF (DebtRecoveryRates[2])	0x0403	0b110 (do not collect when supply is disabled due to no credit)	0b00000	0x020309060000130A01FF09060000130202FF0403C0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130205FF (SecondaryTariffTOUPriceMatrix (Twin element ESME only))	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130205FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130205FF (SecondaryTariffTOUPriceMatrix (Twin element ESME only))	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130205FF0403E0

5605 When the mode is set as in Table 18.2.1.3i:

Payment Mode	Suspend Debt Emergency	Suspend Debt Disabled
Prepayment	True	False

5606 Table 18.2.1.3i: Prepayment states

5607 the *credit_charge_configuration_element* array entries are as per Table 18.2.1.3j.

Tag & Length	Credit object	Tag & Length	Charge object	Tag & Length	Collection bit string	trailing_bits	Array entry
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130200FF (TariffBlockPriceMatrixTOU)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130200FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130204FF (StandingCharge)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130204FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130201FF (DebtRecoveryRates[1])	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130201FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130202FF (DebtRecoveryRates[2])	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130202FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130203FF (DebtRecoveryPerPayment)	0x0403	0b1110 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130203FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130203FF (DebtRecoveryPerPayment)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130203FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130200FF (TariffBlockPriceMatrixTOU)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130200FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130204FF (StandingCharge)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130204FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130201FF (DebtRecoveryRates[1])	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130201FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130202FF (DebtRecoveryRates[2])	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130202FF0403E0
0x02030906	0x0000130A02FF (AccumulatedDebt)	0x0906	0x0000130204FF (StandingCharge)	0x0403	0b111 (collect in Emergency)	0b00000	0x020309060000130A02FF09060000130204FF0403E0

<i>Tag & Length</i>	<i>Credit object</i>	<i>Tag & Length</i>	<i>Charge object</i>	<i>Tag & Length</i>	<i>Collection bit string</i>	<i>trailing_bits</i>	<i>Array entry</i>
					Credit period – see note at bottom of table)		
0x02030906	0x0000130A02FF (AccumulatedDebt)	0x0906	0x0000130201FF (DebtRecoveryRates[1])	0x0403	0b111 (collect in Emergency Credit period – see note at bottom of table)	0b00000	0x020309060000130A02FF09060000130201FF0403E0
0x02030906	0x0000130A02FF (AccumulatedDebt)	0x0906	0x0000130202FF (DebtRecoveryRates[2])	0x0403	0b111 (collect in Emergency Credit period – see note at bottom of table)	0b00000	0x020309060000130A02FF09060000130202FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130205FF (SecondaryTariffTOUPPriceMatrix (Twin element ESME only))	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130205FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130205FF (SecondaryTariffTOUPPriceMatrix (Twin element ESME only))	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130205FF0403E0

5608 Table 18.2.1.3j: *credit_charge_configuration_element* array entries

5609 Note that, as per SMETS, charges shall only accrue to AccumulatedDebt in Emergency Credit periods.

5610 When the mode is set as in Table 18.2.1.3k:

<i>Payment Mode</i>	<i>Suspend Debt Emergency</i>	<i>Suspend Debt Disabled</i>
Prepayment	True	True

5611 the *credit_charge_configuration_element* array entries are as per Table 18.2.1.3l.

<i>Tag & Length</i>	<i>Credit object</i>	<i>Tag & Length</i>	<i>Charge object</i>	<i>Tag & Length</i>	<i>Collection bit string</i>	<i>trailing_bits</i>	<i>Array entry</i>
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130200FF (TariffBlockPriceMatrixTOU)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130200FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130204FF (StandingCharge)	0x0403	0b111 (collectable in all	0b00000	0x020309060000130A00FF09060000130204FF0403E0

Tag & Length	Credit object	Tag & Length	Charge object	Tag & Length	Collection bit string	trailing_bits	Array entry
					circumstances)		
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130201FF (DebtRecoveryRates[1])	0x0403	0b110 (do not collect when supply is disabled due to no credit)	0b00000	0x020309060000130A00FF09060000130201FF0403C0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130202FF (DebtRecoveryRates[2])	0x0403	0b110 (do not collect when supply is disabled due to no credit)	0b00000	0x020309060000130A00FF09060000130202FF0403C0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130203FF (DebtRecoveryPerPayment)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130203FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130203FF (DebtRecoveryPerPayment)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130203FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130200FF (TariffBlockPriceMatrixTOU)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130200FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130204FF (StandingCharge)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130204FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130201FF (DebtRecoveryRates[1])	0x0403	0b110 (do not collect when supply is disabled due to no credit)	0b00000	0x020309060000130A01FF09060000130201FF0403C0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130202FF (DebtRecoveryRates[2])	0x0403	0b110 (do not collect when supply is disabled due to no credit)	0b00000	0x020309060000130A01FF09060000130202FF0403C0
0x02030906	0x0000130A02FF (AccumulatedDebt)	0x0906	0x0000130204FF (StandingCharge)	0x0403	0b111 (collect in Emergency Credit period – see note at bottom of table)	0b00000	0x020309060000130A02FF09060000130204FF0403E0
0x02030906	0x0000130A02FF (AccumulatedDebt)	0x0906	0x0000130201FF (DebtRecoveryRates[1])	0x0403	0b111 (collect in Emergency)	0b00000	0x020309060000130A02FF09060000130201FF0403E0

<i>Tag & Length</i>	<i>Credit object</i>	<i>Tag & Length</i>	<i>Charge object</i>	<i>Tag & Length</i>	<i>Collection bit string</i>	<i>trailing_bits</i>	<i>Array entry</i>
					Credit period – see note at bottom of table)		
0x02030906	0x0000130A02FF (AccumulatedDebt)	0x0906	0x0000130202FF (DebtRecoveryRates[2])	0x0403	0b111 (collect in Emergency Credit period – see note at bottom of table)	0b00000	0x020309060000130A02FF09060000130202FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130205FF (SecondaryTariffTOUPriceMatrix (Twin element ESME only))	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130205FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130205FF (SecondaryTariffTOUPriceMatrix (Twin element ESME only))	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130205FF0403E0

5612

Table 18.2.1.3l: *credit_charge_configuration_element* array entries

5613

Note that, as per SMETS, charges shall only accrue to AccumulatedDebt in Emergency Credit periods.

5614

18.2.1.4 Encoding of Billing Calendar start date-time and periodicity

5615

Table 18.2.1.4 sets out how the components of the Billing Calendar start date-time and periodicity should be encoded.

<i>Component</i>	<i>Hex value</i>	<i>Length in octets</i>	<i>Notes</i>
execution_time			
Tag	0x01	1	Tag for array
Length	0x01	1	1 entry in array
execution_time_date			
Tag	0x02	1	Tag for structure
Length	0x02	1	2 elements in structure
Time			
Tag	0x09	1	Tag for structure
Length	0x04	1	4 octets in DLMS encoded time
Value	See note	4	Time part of the start date-time, as per section 4.1.6.1 of the Blue Book

Component	Hex value	Length in octets	Notes
Date			
Tag	0x09	1	5 octets in DLMS encoded date
Length	0x05	1	2 elements in structure
Value			
year highbyte,	0xFF	1	0xFF means not specified
year lowbyte,	0xFF	1	0xFF means not specified
month,	0xFF	1	0xFF means not specified
day of month,	0xFF unless periodicity is monthly. If periodicity is monthly, this shall be the day of the month of the start date-time.	1	0xFF means not specified.
day of week	0xFF unless periodicity is weekly If periodicity is weekly, this shall be the day of the week of the start date-time.	1	0xFF means not specified

5616 Table 18.2.1.4: Encoding of Billing Calendar start date-time and periodicity

5617

18.3 Illustrative command and response instantiation and DER encoding

5618

18.3.1 Illustrative @UpdateSecurityCredentials.Command instantiation and its DER encoding - informative

5619 supplierUpdatingAllSupplierCertificates in Table 18.3.1a is an ASN.1 structured value assignment. This specific example is
5620 where a Device's Supplier is instructing the Device to replace both the Supplier Digital Signing and Key Agreement credentials on the Device,
5621 and resetting Protection Against Replay counters. In business terms, an example of this would be at Change of Supplier.5622 The black text specifies the parts of the ASN.1 structure, the blue text specifies the value it is set to and the comments explain each of the
5623 values.

ASN.1	Notes
supplierUpdatingAllSupplierCertificates Command ::= {authorisingRemotePartyControl}	This message is for the Supplier replacing supplier

ASN.1	Notes
{credentialsReplacementMode authorisingRemotePartyTACellIdentifier {trustAnchorCellRemotePartyRole supplier, trustAnchorCellKeyUsage { digitalSignature}}},	credentials The public key to be used to check the signature on this message is the supplier digital signing key currently held by the Device.
authorisingRemotePartySeqNumber 123456789,	This is the existing supplier's counter, so greater than any this supplier has used
newRemotePartyFloorSeqNumber 987654321}	This is the new supplier's counter, which the Device should use if the Command is successful
replacements {replacementCertificate '0A7C8E9F123456789ABCDEF01234'H, targetTrustAnchorCell {trustAnchorCellRemotePartyRole supplier, trustAnchorCellKeyUsage { digitalSignature}}} {replacementCertificate '0B34269F123456789ABCDEF01234'H, targetTrustAnchorCell {trustAnchorCellRemotePartyRole supplier, trustAnchorCellKeyUsage {keyAgreement}}}}	The new supplier's digital signing certificate which is to be placed in the Device's supplier, digital signature Trust Anchor Cell
certificationPathCertificates }}	The new supplier's key agreement certificate... which is to be placed in the Device's supplier, key agreement Trust Anchor Cell The Certificate for the CA which issued the new supplier's certificates. The Device will use this to check that the new supplier certificates were properly issued.

- 5624 Table 18.3.1a: Illustrative @UpdateSecurityCredentials.Command instantiation – ASN.1 structure
- 5625 The message sent to the Device would contain the DER encoding of the above ASN.1 value assignment. This DER encoding is laid out and
5626 explained in Table 18.3.1b. For these purpose, the Certificate is simply shown as an OCTET STRING.

5627

Component	Value	Notes
Payload SEQUENCE:		
tag = [UNIVERSAL 16] constructed;	0x30	Tag for SEQUENCE
length =	0x64	100 octet length follows
contents =:		
authorisingRemotePartyControl AuthorisingRemotePartyControl SEQUENCE:		
tag = [UNIVERSAL 16] constructed;	0x30	Tag for SEQUENCE
length =	0x18	Length of authorisingRemotePartyControl
contents =:		
credentialsReplacementMode CredentialsReplacementMode INTEGER:		
tag = [UNIVERSAL 2] primitive;	0x02	
length =	0x01	
contents =:	0x02	Representing supplierBySupplier
authorisingRemotePartyTACellIdentifier TrustAnchorCellIdentifier SEQUENCE:		
tag = [2] constructed;	0xA2	Tag for authorisingRemotePartyTACellIdentifier
length =	0x07	Length of authorisingRemotePartyTACellIdentifier
contents =:		
trustAnchorCellRemotePartyRole RemotePartyRole INTEGER:		
tag = [UNIVERSAL 2] primitive;	0x02	Tag for INTEGER
length =	0x01	1 octet length INTEGER
contents =:	0x02	Representing supplier RemotePartyRole
trustAnchorCellKeyUsage KeyUsage BIT STRING:		
tag = [UNIVERSAL 3] primitive;	0x03	Tag for BIT STRING
length =	0x02	2 octet length BIT STRING
contents =:	0x0780	Representing digitalSignature
authorisingRemotePartySeqNumber SeqNumber INTEGER:		

Component	Value	Notes
tag = [4] primitive;	0x84	Tag for INTEGER
length =	0x04	4 octet length INTEGER
contents =:	0x075bcd15	The old supplier's Protection Against Replay counter in hex
newRemotePartyFloorSeqNumber SeqNumber INTEGER:		
tag = [5] primitive;	0x85	Tag for INTEGER
length =	0x04	4 octet length INTEGER
contents =:	0x3ade68b1	The new supplier's Protection Against Replay counter in hex
replacements SEQUENCE OF:		
tag = [UNIVERSAL 16] constructed;	0x30	Tag for SEQUENCE
length =	0x38	Length of replacements
contents =:		
TrustAnchorReplacement SEQUENCE:		
tag = [UNIVERSAL 16] constructed;	0x30	Tag for SEQUENCE
length =	0x19	Length of first TrustAnchorReplacement
contents =:		
replacementCertificate Certificate OCTET STRING:		
tag = [UNIVERSAL 4] primitive;	0x04	Tag for OCTET STRING
length =	0x0e	Length of certificate
contents =:	0x0a7c8e9f123456789abcdef01234	New supplier's digitalSignature certificate
targetTrustAnchorCell TrustAnchorCellIdentifier SEQUENCE:		
tag = [UNIVERSAL 16] constructed;	0x30	Tag for SEQUENCE
length =	0x07	Length of targetTrustAnchorCell
contents =:		
trustAnchorCellRemotePartyRole RemotePartyRole INTEGER:		
tag = [UNIVERSAL 2] primitive;	0x02	Tag for INTEGER
length =	0x01	1 octet length INTEGER
contents =:	0x02	Representing supplier RemotePartyRole

Component	Value	Notes
trustAnchorCellKeyUsage KeyUsage BIT STRING:		
tag = [UNIVERSAL 3] primitive;	0x03	Tag for BIT STRING
length =	0x02	2 octet length BIT STRING
contents =:	0x0780	Representing digitalSignature
TrustAnchorReplacement SEQUENCE:		
tag = [UNIVERSAL 16] constructed;	0x30	Tag for SEQUENCE
length =	0x19	Length of second TrustAnchorReplacement
contents =:		
replacementCertificate Certificate OCTET STRING:		
tag = [UNIVERSAL 4] primitive;	0x04	Tag for OCTET STRING
length =	0x0e	Length of certificate
contents =:	0x0b34269f123456789abcdef01234	New supplier's keyAgreement certificate
targetTrustAnchorCell TrustAnchorCellIdentifier SEQUENCE:		
tag = [UNIVERSAL 16] constructed;	0x30	Tag for SEQUENCE
length =	0x07	Length of targetTrustAnchorCell
contents =:		
trustAnchorCellRemotePartyRole RemotePartyRole INTEGER:		
tag = [UNIVERSAL 2] primitive;	0x02	Tag for INTEGER
length =	0x01	1 octet length INTEGER
contents =:	0x02	Representing supplier RemotePartyRole
trustAnchorCellKeyUsage KeyUsage BIT STRING:		
tag = [UNIVERSAL 3] primitive;	0x03	Tag for BIT STRING
length =	0x02	2 octet length BIT STRING
contents =:	0x0308	Representing keyAgreement
certificationPathCertificates SEQUENCE OF:		
tag = [UNIVERSAL 16] constructed;	0x30	Tag for SEQUENCE
length =	0x10	Length of certificationPathCertificates
contents =:		
Certificate OCTET STRING:		
tag = [UNIVERSAL 4] primitive;	0x04	Tag for OCTET STRING

Component	Value	Notes
length =	0x0e	Length of certificate
contents =:	0xfffaabb9f123456789abcdef01234	CA certificate for new supplier

Table 18.3.1b: Illustrative @UpdateSecurityCredentials.Command instantiation – DER encoding

5628 **18.3.2 Illustrative @UpdateSecurityCredentials.Response instantiation and its DER encoding -**
 5629 **informative**

5630 supplierUpdatingAllSupplierCertificatesResponse in Table 18.3.2a is an ASN.1 structured value assignment. This specific
 5631 example is where a Device is responding successfully to a Command.

5632 The black text specifies the parts of the ASN.1 structure, the *blue text* specifies the value it is set to by the Device and the comments explain
 5633 each of the values.

ASN.1	Notes
<pre> supplierUpdatingAllSupplierCertificatesResponse Response ::= {credentialsReplacementMode <i>supplierBySupplier</i>, remotePartySeqNumberChanges {{otherRemotePartyRole <i>supplier</i>, otherRemotePartyFloorSeqNumber 987654321} }, } replacementOutcomes { {affectedTrustAnchorCell trustAnchorCellRemotePartyRole <i>supplier</i>, trustAnchorCellKeyUsage { <i>digitalSignature</i>}, statusCode success, existingSubjectUniqueID '123456789ABCDEF0'H, } </pre>	<p>The corresponding Command was for the Supplier replacing supplier credentials</p> <p>This is the new supplier's counter, which the Device will now use for Protection Against Replay in relation to the supplier role</p>
	<p>This outcome is for the supplier digital signing store</p> <p>The old supplier's Entity Identifier</p> <p>The KeyIdentifier for the old supplier's digital signing</p>

ASN.1	Notes
<pre> existingSubjectKeyIdentifier '1234567890123456'H, replacingSubjectUniqueID 'FEDCBA9876543210'H, replacingSubjectKeyIdentifier 'ABCDEABCDEABCDEA'H}, {affectedTrustAnchorCell {trustAnchorCellRemotePartyRole supplier, trustAnchorCellKeyUsage { statusCode existingSubjectUniqueID '123456789ABCDEF0'H, existingSubjectKeyIdentifier '0987654321098765'H, replacingSubjectUniqueID 'FEDCBA9876543210'H, replacingSubjectKeyIdentifier 'FEDCBFEDCBFEDCBF'H}}} </pre>	<p>key The new supplier's Entity Identifier The KeyIdentifier for the old supplier's digital signing key KeyIdentifier for the new supplier's digital signing key This outcome is for the supplier key agreement store</p>

5634 Table 18.3.2a: Illustrative @UpdateSecurityCredentials.Response instantiation – ASN.1 structure

5635 The message sent by the Device would contain the DER encoding of the above ASN.1 value assignment. This DER encoding is laid out and
5636 explained in Table 18.3.2b.

Component	Value	Notes
Response SEQUENCE:		
tag = [UNIVERSAL 16] constructed;	0X30	Tag for SEQUENCE
length =	0X7E	Length 126
content =		
credentialsReplacementMode CredentialsReplacementMode INTEGER:		
tag = [UNIVERSAL 2] primitive;	0X02	Tag for INTEGER
length =	0X01	
content =	0X02	Value for supplierBySupplier
remotePartySeqNumberChanges SEQUENCE OF:		
tag = [UNIVERSAL 16] constructed;	0X30	Tag for SEQUENCE
length =	0X0B	
content =		

Component	Value	Notes
RemotePartySeqNumberChange SEQUENCE:		
tag = [UNIVERSAL 16] constructed;	0X30	Tag for SEQUENCE
length =	0X09	
content =		
otherRemotePartyRole RemotePartyRole INTEGER:		
tag = [UNIVERSAL 2] primitive;	0X02	Tag for INTEGER
length =	0X01	
content =	0X02	Value for supplier
otherRemotePartyFloorSeqNumber SeqNumber INTEGER:		
tag = [UNIVERSAL 2] primitive;	0X02	Tag for INTEGER
length =	0X04	
content =	0X3ADE68B1	The new supplier's Protection Against Replay counter in hexadecimal
replacementOutcomes SEQUENCE OF:		
tag = [UNIVERSAL 16] constructed;	0X30	Tag for SEQUENCE
length =	0X6C	Length of 108
content =		
ReplacementOutcome SEQUENCE:		
tag = [UNIVERSAL 16] constructed;	0X30	Tag for SEQUENCE
length =	0X34	Length of 52
content =		
affectedTrustAnchorCell TrustAnchorCellIdentifier SEQUENCE:		
tag = [UNIVERSAL 16] constructed;	0X30	Tag for SEQUENCE
length =	0X07	
content =		
trustAnchorCellRemotePartyRole RemotePartyRole INTEGER:		
tag = [UNIVERSAL 2] primitive;	0X02	Tag for INTEGER
length =	0X01	
content =	0X02	Value for supplier
trustAnchorCellKeyUsage KeyUsage BIT STRING:		

Component	Value	Notes
tag = [UNIVERSAL 3] primitive;	0x03	Tag for BIT STRING
length =	0X02	
content =	0X0780	Tag for digitalSignature
statusCode StatusCode ENUMERATED:		
tag = [UNIVERSAL 10] primitive;	0X0A	Tag for ENUMERATED
length =	0X01	
content =	0X00	Value for success
existingSubjectUniqueID OCTET STRING:		
tag = [UNIVERSAL 4] primitive;	0X04	Tag for OCTET STRING
length =	0X08	8 octet length of Entity Identifier
content =	0X123456789ABCDEF0	
existingSubjectKeyIdentifier OCTET STRING:		
tag = [UNIVERSAL 4] primitive;	0X04	Tag for OCTET STRING
length =	0X08	length of KeyIdentifier
content =	0X1234567890123456	KeyIdentifier
replacingSubjectUniqueID OCTET STRING:		
tag = [UNIVERSAL 4] primitive;	0X04	Tag for OCTET STRING
length =	0X08	8 octet length of Entity Identifier
content =	0XFEDCBA9876543210	
replacingSubjectKeyIdentifier OCTET STRING:		
tag = [UNIVERSAL 4] primitive;	0X04	Tag for OCTET STRING
length =	0X08	length of KeyIdentifier
content =	0XABCDEABCDEABCDEA	KeyIdentifier
ReplacementOutcome SEQUENCE:		
tag = [UNIVERSAL 16] constructed;	0X30	Tag for SEQUENCE
length =	0X34	
content =		
affectedTrustAnchorCell TrustAnchorCellIdentifier SEQUENCE:		
tag = [UNIVERSAL 16] constructed;	0X30	Tag for SEQUENCE
length =	0X07	

Component	Value	Notes
content =		
trustAnchorCellRemotePartyRole RemotePartyRole INTEGER:		
tag = [UNIVERSAL 2] primitive;	0X02	Tag for INTEGER
length =	0X01	
content =	0X02	Value for supplier
trustAnchorCellKeyUsage KeyUsage BIT STRING:		
tag = [UNIVERSAL 3] primitive;	0x03	Tag for BIT STRING
length =	0X02	
content =	0X0308	
statusCode StatusCode ENUMERATED:		
tag = [UNIVERSAL 10] primitive;	0XA	Tag for ENUMERATED
length =	0X01	
content =	0X00	Value for success
existingSubjectUniqueID OCTET STRING:		
tag = [UNIVERSAL 4] primitive;	0X04	Tag for OCTET STRING
length =	0X08	8 octet length of Entity Identifier
content =	0X123456789ABCDEF0	
existingSubjectKeyIdentifier OCTET STRING:		
tag = [UNIVERSAL 4] primitive;	0X04	Tag for OCTET STRING
length =	0X08	length of KeyIdentifier
content =	0X0987654321098765	KeyIdentifier
replacingSubjectUniqueID OCTET STRING:		
tag = [UNIVERSAL 4] primitive;	0X04	Tag for OCTET STRING
length =	0X08	8 octet length of Entity Identifier
content =	0XFEDCBA9876543210	
replacingSubjectKeyIdentifier OCTET STRING:		
tag = [UNIVERSAL 4] primitive;	0X04	Tag for OCTET STRING
length =	0X08	length of KeyIdentifier
content =	0XFEDCBFEDCBFEDCBF	KeyIdentifier

Table 18.3.2b: Illustrative @UpdateSecurityCredentials.Response instantiation – DER encoding

5637

18.4 Cryptographic Test Vectors

5638
5639
5640
5641

This Section 18.4 provides cryptographic calculations in relation to a number of sample messages. The sample messages' contents align with the corresponding Message Templates in Section 18.2. To undertake cryptographic calculations, a number of details about the Smart Metering Entities involved are also required, not least Key Pairs, Entity Identifiers and Originator Counters. This section specifies and uses sample values of such attributes.

5642

18.4.1 Cryptographic Calculations

5643

Create details for three Smart Metering Entities with associated Keys and shared secrets:

5644

5645

An Entity called SupplierA:

--With an Entity ID:	0x12:34:56:78:9A:BC:DE:FO
--With a current Originator Counter:	0x00:00:00:00:00:00:00:01
--Digital Signing Private key :	0x3A:6B:2E:AA:0D:9F:25:A9:E4:55:98:3F:EB:5B:B9:47:52:81:21:91:1B:F3:B7:6B:E5:66:1C:89:DB:F2:4B:26
--Digital Signing Public key :	0x76:62:8E:1C:84:EF:79:35:54:8A:E5:D6:2C:7B:B3:AD:28:96:4C:F7:94:FO:38:7A:69:7E:EC:19:CD:D9:8F:46:0A:4D:5E:19:08:7E:F7:21:6E:D8:9C:29:83:1A:6E:E8:38:C8:DE:88:EF:34:F1:1D:3F:41:F3:6D:80:B2:A5:D5
--Key Agreement Private key :	0x3D:9D:FB:33:2E:B4:D6:D6:06:D7:47:18:55:3E:5E:61:B3:92:BO:FC:4C:90:CE:6A:A4:CE:DA:81:7E:80:11:B1
--Key Agreement Public key :	0xEF:F2:1D:5D:D6:74:EE:C6:EO:87:40:70:3B:52:25:52:CB:B7:4F:FC:A1:15:36:C5:37:C3:C8:06:E4:14:3C:8F:B2:E7:CA:3E:73:06:CB:46:DB:E4:BD:59:9C:C4:A3:1F:7

	8:8C:2F:B7:A9:B9:BC:97:BE:98:C8:1E:F1:82:1A:30
--The shared secret calculated with DeviceA is :	0x15:45:AD:F2:75:DC:8E:57:AB:E4:71:E9:F0:C1:20:C2:FA:DD:5B:12:51:AF:B7:BD :AB:25:3C:80:1B:41:11:CE

5646

5647 An Entity called Access Control Broker:

--With an Entity ID:	0xAB:AB:AB:AB:AB:AB:AB:AB
--With a current Originator Counter:	0x10:00:00:00:00:00:00:01
--Key Agreement Private key :	0xE4:A6:CF:B4:31:47:1C:FC:AE:49:1F:D5:66:D1:9C:87:08:2C:F9:FA:77:22:D7:FA: 24:B2:B3:F5:66:9D:BE:FB
--Key Agreement Public key :	0x29:2F:97:FE:C1:B3:0C:38:49:B8:06:D9:04:46:E4:AO:37:D6:D1:78:01:97:96:E7 :6E:52:55:BD:C3:AO:8E:34:6F:9F:6E:6E:7E:8F:6A:4D:55:96:2D:2F:2D:OE:16:CF:F2 :7B:F3:F9:25:FA:7D:BA:FD:15:A8:B1:DC:69:58:94
--The shared secret calculated with DeviceA is :	0x9A:AC:F2:E6:D5:1B:D5:FF:8F:37:BF:36:80:19:A6:91:CB:5B:2F:CB:7B:5F:03:OA: 00:06:36:47:B2:OE:13:FE

5648

5649 An Entity called DeviceA:

--With an Entity ID:	0xFF:FF:FF:FF:FF:FF:FF:FE
----------------------	---------------------------

--With a current Originator Counter:	0x20:00:00:00:00:00:00:01
--Digital Signing Private key :	0xFC:9B:B7:73:E6:C8:35:OA:DB:40:51:AC:91:3C:A4:70:CF:42:2D:8A:53:DE:8C:88 :1D:BF:FE:B4:OB:A4:70:51
--Digital Signing Public key :	0x86:FB:5E:B3:CA:05:07:22:6B:E7:19:70:58:B9:EC:04:1D:3A:37:58:D9:D9:C9:19 :02:AC:A3:39:1F:4E:58:AE:F1:3A:FF:63:CC:4E:F6:89:42:B9:B9:49:04:DC:1B:89:0 E:DB:EA:BD:16:B9:92:11:06:24:96:8E:89:4E:56:0E
--Key Agreement Private key :	0xFB:9F:4C:02:B7:AB:F8:B0:DA:BA:02:7E:OB:C8:1B:8D:D2:09:68:3B:1C:88:93:EE :45:3F:AD:F3:A8:0F:73:E5
--Key Agreement Public key :	0x2D:B4:5A:3F:21:88:94:38:B4:2C:8F:46:4C:75:29:2B:AC:F5:FD:DB:5D:AO:B4:92 :50:1B:29:9C:BF:E9:2D:8F:DB:90:FC:8F:F4:02:61:29:83:8B:1B:CA:D1:40:2C:AE:4 7:FE:7D:80:84:E4:09:A4:1A:FC:E1:6D:63:57:9C:5F
--The shared secret calculated with AccessControlBroker is :	0x9A:AC:F2:E6:D5:1B:D5:FF:8F:37:BF:36:80:19:A6:91:CB:5B:2F:CB:7B:5F:03:0A: 00:06:36:47:B2:0E:13:FE
--The shared secret calculated with SupplierA is :	0x15:45:AD:F2:75:DC:8E:57:AB:E4:71:E9:F0:C1:20:C2:FA:DD:5B:12:51:AF:B7:BD :AB:25:3C:80:1B:41:11:CE

5650

5651 Create a Critical Command from SupplierA to Device A: ECS04b Reset Meter Balance on the ESME:

--GBCS Message Category:	SME.C.C
--------------------------	---------

--GBCS Message Type:	Command
--CRA Flag:	0x01
--Originator Counter:	0x00:00:00:00:00:00:00:01
--Business Originator ID:	0x12:34:56:78:9A:BC:DE:FO
--Business Target ID:	0xFF:FF:FF:FF:FF:FF:FF:FE
--Date Time:	0x
--Other Info:	0x00:B3 It is a Message Code for ECS04b reset meter balance on ESME
--Message Content:	0xD9:20:00:00:01:00:03:03:00:70:00:00:13:0A:00:FF:02:03:00:70:00:00:13:0A :01:FF:02:03:00:70:00:00:13:0A:02:FF:02:03:05:00:00:00:05:00:00:00:00: 5:00:00:00:00
--The originator's Private Signing Key:	0x3A:6B:2E:AA:0D:9F:25:A9:E4:55:98:3F:EB:5B:B9:47:52:81:21:91:1B:F3:B7:6B :E5:66:1C:89:DB:F2:4B:26
--The Message parts used in Signing:	0x01:00:00:00:00:00:00:00:01:12:34:56:78:9A:BC:DE:FO:FF:FF:FF:FF:FF:FE: 00:B3:D9:20:00:00:01:00:03:03:00:70:00:00:13:0A:00:FF:02:03:00:70:00:00:1 3:0A:01:FF:02:03:00:70:00:00:13:0A:02:FF:02:03:05:00:00:00:05:00:00:00: 00:05:00:00:00:00
--The per message secret number:	28321578986444545792209120900555608833352738719916097837081

	350912149044905275
--The resulting Signature in Plain Format:	0x85:AE:39:D4:5D:5C:73:A4:40:70:DF:71:C7:AO:97:6B:AF:60:A3:62:6E:6D:08:D 1:67:AA:7C:F4:AB:83:93:BO:B4:13:E9:1D:3E:79:FD:6C:CC:93:F4:5D:BO:A2:OB:E5: 26:4B:5C:E9:BA:56:A2:47:00:72:78:4D:D1:A1:17:52
--The Grouping Header:	0xDF:09:01:00:00:00:00:00:00:01:08:12:34:56:78:9A:BC:DE:F0:08:FF:FF:FF: F:FF:FF:FF:FE:00:02:00:B3:35
--All of the Message parts covered by the general-signing structure	0xDF:09:01:00:00:00:00:00:00:00:01:08:12:34:56:78:9A:BC:DE:F0:08:FF:FF:FF: F:FF:FF:FF:FE:00:02:00:B3:35:D9:20:00:00:01:00:03:03:00:70:00:00:13:0A:00:F F:02:03:00:70:00:00:13:0A:01:FF:02:03:00:70:00:00:13:0A:02:FF:02:03:05:00: 00:00:05:00:00:00:00:05:00:00:00:00:40:85:AE:39:D4:5D:5C:73:A4:40:70:D F:71:C7:AO:97:6B:AF:60:A3:62:6E:6D:08:D1:67:AA:7C:F4:AB:83:93:BO:B4:13:E9: 1D:3E:79:FD:6C:CC:93:F4:5D:BO:A2:OB:E5:26:4B:5C:E9:BA:56:A2:47:00:72:78:4 D:D1:A1:17:52
--The KDF OtherInfo:	0x60:85:74:06:08:03:00:12:34:56:78:9A:BC:DE:F0:09:01:00:00:00:00:00: 01:FF:FF:FF:FF:FF:FF:FE
--The per message secret symmetric key:	177594815140134193685548970760141301611
--The Initialization Vector:	0x12:34:56:78:9A:BC:DE:F0:00:00:00:00
--The Additional Authenticated Data:	0x11:DF:09:01:00:00:00:00:00:00:01:08:12:34:56:78:9A:BC:DE:F0:08:FF:FF: FF:FF:FF:FF:FE:00:02:00:B3:35:D9:20:00:00:01:00:03:03:00:70:00:00:13:0A:

	00:FF:02:03:00:70:00:00:13:0A:01:FF:02:03:00:70:00:00:13:0A:02:FF:02:03:05 :00:00:00:05:00:00:00:00:05:00:00:00:00:40:85:AE:39:D4:5D:5C:73:A4:40: 70:DF:71:C7:A0:97:6B:AF:60:A3:62:6E:6D:08:D1:67:AA:7C:F4:AB:83:93:B0:B4:1 3:E9:1D:3E:79:FD:6C:CC:93:F4:5D:B0:A2:0B:E5:26:4B:5C:E9:BA:56:A2:47:00:72: 78:4D:D1:A1:17:52
--The resulting MAC:	0x43:0C:DE:EA:CC:82:97:09:44:71:CF:92
--The MAC Header excluding the Security Header	0xDD:00:00:00:00:00:00:82:00:A9
--The Security Header fields:	0x11:00:00:00:00
--The resulting Message:	0xDD:00:00:00:00:00:00:82:00:A9:11:00:00:00:00:DF:09:01:00:00:00:00:00: 0:00:01:08:12:34:56:78:9A:BC:DE:F0:08:FF:FF:FF:FF:FF:FF:FE:00:02:00:B3:35: D9:20:00:00:01:00:03:03:00:70:00:00:13:0A:00:FF:02:03:00:70:00:00:13:0A:0 1:FF:02:03:00:70:00:00:13:0A:02:FF:02:03:05:00:00:00:00:05:00:00:00:05: 00:00:00:40:85:AE:39:D4:5D:5C:73:A4:40:70:DF:71:C7:A0:97:6B:AF:60:A3:6 2:6E:6D:08:D1:67:AA:7C:F4:AB:83:93:B0:B4:13:E9:1D:3E:79:FD:6C:CC:93:F4:5D: B0:A2:0B:E5:26:4B:5C:E9:BA:56:A2:47:00:72:78:4D:D1:A1:17:52:43:0C:DE:EA:C C:82:97:09:44:71:CF:92

5652

5653 And get a Critical Response to SupplierA from Device A: ECS04b Reset Meter Balance on the ESME:

--GBCS Message Category:	SME.C.C
--------------------------	---------

--GBCS Message Type:	Response
--CRA Flag:	0x02
--Originator Counter:	0x00:00:00:00:00:00:00:01
--Business Originator ID:	0xFF:FF:FF:FF:FF:FF:FF:FE
--Business Target ID:	0x12:34:56:78:9A:BC:DE:FO
--Date Time:	0x
--Other Info:	0x00:B3
--Message Content:	0xDA:20:00:00:01:00:00:03:00:00:00:03:03:00:03:00:03:00
--The originator's Private Signing Key:	0xFC:9B:B7:73:E6:C8:35:0A:DB:40:51:AC:91:3C:A4:70:CF:42:2D:8A:53:DE:8C:88 :1D:BF:FE:B4:0B:A4:70:51
--The Message parts used in Signing:	0x02:00:00:00:00:00:00:00:00:01:FF:FF:FF:FF:FF:FF:FE:12:34:56:78:9A:BC:DE:FO: 00:B3:DA:20:00:00:01:00:00:03:00:00:00:03:03:00:03:00:03:00
--The per message secret number:	48814838122802850934537136292612629832407092209107840231664 691912455948374928
--The resulting Signature in Plain Format:	0x01:99:E9:84:CE:C7:5D:DC:A7:F1:DD:F6:E5:3E:2E:67:35:2A:2B:E3:8A:4B:66:F8: ED:59:66:06:FA:B9:83:FF:30:0C:AA:76:DE:88:CE:D9:D5:63:A5:C0:3E:8F:3A:7C:0

	0:07:80:F3:F2:06:1C:61:1E:9A:AO:B1:8B:46:OD:77
--The Grouping Header:	0xDF:09:02:00:00:00:00:00:00:00:01:08:FF:FF:FF:FF:FF:FF:FE:08:12:34:56:78: 9A:BC:DE:F0:00:02:00:B3:12
--All of the Message parts covered by the general-signing structure	0xDF:09:02:00:00:00:00:00:00:00:01:08:FF:FF:FF:FF:FF:FF:FE:08:12:34:56:78: 9A:BC:DE:F0:00:02:00:B3:12:DA:20:00:00:01:00:00:03:00:00:00:03:03:00:03:0 0:03:00:40:01:99:E9:84:CE:C7:5D:DC:A7:F1:DD:F6:E5:3E:2E:67:35:2A:2B:E3:8A: 4B:66:F8:ED:59:66:06:FA:B9:83:FF:30:0C:AA:76:DE:88:CE:D9:D5:63:A5:CO:3E:8F :3A:7C:00:07:80:F3:F2:06:1C:61:1E:9A:AO:B1:8B:46:OD:77
--The resulting Message:	0xDF:09:02:00:00:00:00:00:00:00:00:01:08:FF:FF:FF:FF:FF:FF:FE:08:12:34:56:78: 9A:BC:DE:F0:00:02:00:B3:12:DA:20:00:00:01:00:00:03:00:00:00:00:03:03:00:03:0 0:03:00:40:01:99:E9:84:CE:C7:5D:DC:A7:F1:DD:F6:E5:3E:2E:67:35:2A:2B:E3:8A: 4B:66:F8:ED:59:66:06:FA:B9:83:FF:30:0C:AA:76:DE:88:CE:D9:D5:63:A5:CO:3E:8F :3A:7C:00:07:80:F3:F2:06:1C:61:1E:9A:AO:B1:8B:46:OD:77

5654

5655 Supplier A has now increased its Originator Counter by 1.

5656

5657 Create a non-Critical Command from SupplierA to Device A: ECS12 Set Change of Tenancy date on ESME:

--GBCS Message Category:	SME.C.NC
--GBCS Message Type:	Command

--CRA Flag:	0x01
--Originator Counter:	0x00:00:00:00:00:00:00:02
--Business Originator ID:	0x12:34:56:78:9A:BC:DE:FO
--Business Target ID:	0xFF:FF:FF:FF:FF:FF:FF:FE
--Date Time:	0x
--Other Info:	0x00:22
--Message Content:	0xD9:20:00:00:02:00:01:02:00:01:00:00:5E:2C:03:02:02:01:09:0C:07:DF:01:05 :FF:00:00:00:80:00:FF
--The Grouping Header:	0xDF:09:01:00:00:00:00:00:00:00:02:08:12:34:56:78:9A:BC:DE:FO:08:FF:FF:FF: F:FF:FF:FF:FE:00:02:00:22:20
--All of the Message parts covered by the general-signing structure	0xDF:09:01:00:00:00:00:00:00:00:00:02:08:12:34:56:78:9A:BC:DE:FO:08:FF:FF:FF: F:FF:FF:FF:FE:00:02:00:22:20:D9:20:00:00:02:00:01:02:00:01:00:00:5E:2C:03:0 2:02:01:09:0C:07:DF:01:05:FF:00:00:00:80:00:FF:00
--The KDF OtherInfo:	0x60:85:74:06:08:03:00:12:34:56:78:9A:BC:DE:FO:09:01:00:00:00:00:00:00 :02:FF:FF:FF:FF:FF:FF:FE
--The per message secret symmetric key:	323267885984686097664772256155520506945

--The Initialization Vector:	0x12:34:56:78:9A:BC:DE:F0:00:00:00:00
--The Additional Authenticated Data:	0x11:DF:09:01:00:00:00:00:00:00:02:08:12:34:56:78:9A:BC:DE:F0:08:FF:FF: FF:FF:FF:FF:FF:FE:00:02:00:22:20:D9:20:00:00:02:00:01:02:00:01:00:00:5E:2C:0 3:02:02:01:09:0C:07:DF:01:05:FF:00:00:00:00:80:00:FF:00
--The resulting MAC:	0xD7:48:D3:F8:7C:97:64:E4:2D:68:1C:11
--The MAC Header excluding the Security Header	0xDD:00:00:00:00:00:00:54
--The Security Header fields:	0x11:00:00:00:00
--The resulting Message:	0xDD:00:00:00:00:00:00:54:11:00:00:00:00:DF:09:01:00:00:00:00:00:00:00: 2:08:12:34:56:78:9A:BC:DE:F0:08:FF:FF:FF:FF:FF:FE:00:02:00:22:20:D9:20: 00:00:02:00:01:02:00:01:00:00:5E:2C:03:02:02:01:09:0C:07:DF:01:05:FF:00:0 0:00:00:80:00:FF:00:D7:48:D3:F8:7C:97:64:E4:2D:68:1C:11

5658

5659 And get a non-Critical Response to SupplierA from Device A: ECS12 Set Change of Tenancy date on ESME:

--GBCS Message Category:	SME.C.NC
--GBCS Message Type:	Response
--CRA Flag:	0x02

--Originator Counter:	0x00:00:00:00:00:00:00:02
--Business Originator ID:	0xFF:FF:FF:FF:FF:FF:FF:FE
--Business Target ID:	0x12:34:56:78:9A:BC:DE:F0
--Date Time:	0x
--Other Info:	0x00:22
--Message Content:	0xDA:20:00:00:02:00:00:01:00:01:02:00
--The Grouping Header:	0xDF:09:02:00:00:00:00:00:00:00:02:08:FF:FF:FF:FF:FF:FF:FE:08:12:34:56:78: 9A:BC:DE:F0:00:02:00:22:0C
--All of the Message parts covered by the general-signing structure	0xDF:09:02:00:00:00:00:00:00:00:00:02:08:FF:FF:FF:FF:FF:FF:FE:08:12:34:56:78: 9A:BC:DE:F0:00:02:00:22:0C:DA:20:00:00:02:00:00:01:00:01:02:00:00
--The KDF OtherInfo:	0x60:85:74:06:08:03:00:FF:FF:FF:FF:FF:FF:FE:09:02:00:00:00:00:00:00:02: 12:34:56:78:9A:BC:DE:F0
--The per message secret symmetric key:	102613665902023293907968102748610736248
--The Initialization Vector:	0xFF:FF:FF:FF:FF:FF:FF:FE:00:00:00:00
--The Additional Authenticated Data:	0x11:DF:09:02:00:00:00:00:00:00:00:02:08:FF:FF:FF:FF:FF:FF:FE:08:12:34:56: 78:9A:BC:DE:F0:00:02:00:22:0C:DA:20:00:00:02:00:00:01:00:01:02:00:00

5660

18.4.2 Example Messages Produced

5661

ECS04b Reset Meter Balance on the ESME (Message Category: SME.C.C)

Command Message Structure		
Name	Encoded Content	Encoded Length
MAC Header (general-ciphering)		
tag	0xDD	1
contents	0x0000000000000000	6
ciphered-service		
length	0x8200A9	3
security header		
security control byte (SC)	0x11	1

invocation counter (IC)	0x00000000	4
Grouping Header (general-signing)		
tag	0xDF	1
transaction-id		
length	0x09	1
value (CRA FLAG)	0x01	1
value (Originator Counter)	0x0000000000000001	8
originator-system-title		
length	0x08	1
value	0x123456789ABCDEFO	8
recipient-system-title		
length	0x08	1
value	0xFFFFFFFFFFFFFFFE	8
date-time		
length	0x00	1
other-information		
Length	0x02	1
Message Code	0x00B3	2
content		
length	0x35	1
access-request		
tag	0xD9	1
long-invoke-id-and-priority		
configuration	0x20	1

<u>invoke-id</u>	0x000001	3
<u>date-time</u>	0x00	1
<u>access-request-body</u>		
<u>access-request-specification</u>		
<u>SEQUENCE OF</u>	0x03	1
<u>Request number 1</u>		
<u> access-request-action</u>	0x03	1
<u> cosem-method-descriptor</u>		
<u> class-id</u>	0x0070	2
<u> instance-id</u>	0x0000130A00FF	6
<u> method-id</u>	0x02	1
<u>Request number 2</u>		
<u> access-request-action</u>	0x03	1
<u> cosem-method-descriptor</u>		
<u> class-id</u>	0x0070	2
<u> instance-id</u>	0x0000130A01FF	6
<u> method-id</u>	0x02	1
<u>Request number 3</u>		
<u> access-request-action</u>	0x03	1
<u> cosem-method-descriptor</u>		
<u> class-id</u>	0x0070	2
<u> instance-id</u>	0x0000130A02FF	6
<u> method-id</u>	0x02	1
<u> access-request-list-of-data</u>		

<u>SEQUENCE OF</u>	<i>0x03</i>	<i>1</i>
<u>Parameter for request number 1</u>		
<u>Names</u>		
<u> Tag</u>	<i>0x05</i>	<i>1</i>
<u> Value</u>	<i>0x00000000</i>	<i>4</i>
<u>Parameter for request number 2</u>		
<u>Names</u>		
<u> Tag</u>	<i>0x05</i>	<i>1</i>
<u> Value</u>	<i>0x00000000</i>	<i>4</i>
<u>Parameter for request number 3</u>		
<u>Names</u>		
<u> Tag</u>	<i>0x05</i>	<i>1</i>
<u> Value</u>	<i>0x00000000</i>	<i>4</i>
<u>signature-length</u>	<i>0x40</i>	<i>1</i>
<u>signature-content</u>	<i>0x85AE39D45D5C73A44070DF71C7A0976BAF60 A3626E6D08D167AA7CF4AB8393B0B413E91D3E 79FD6CCC93F45DB0A20BE5264B5CE9BA56A247 0072784DD1A11752</i>	<i>64</i>
<u>mac-content</u>	<i>0x430CDEEACC8297094471CF92</i>	<i>12</i>

Response Message Structure

Name	Encoded Content	Encoded Length
Grouping Header (general-signing)		
tag	0XDF	1
transaction-id		
length	0X09	1
value (CRA FLAG)	0X02	1
value (Originator Counter)	0X0000000000000001	8
originator-system-title		
length	0X08	1
value	0xFFFFFFFFFFFFFFFE	8
recipient-system-title		
length	0X08	1
value	0X123456789ABCDEFO	8
date-time		
length	0X00	1
other-information		
Length	0X02	1
Message Code	0X00B3	2
content		
length	0X12	1
access-response		
tag	0XDA	1

<u>long-invoke-id-and-priority</u>		
<u> configuration</u>	0x20	1
<u> invoke-id</u>	0x0000001	3
<u> date-time</u>	0x00	1
<u> access-request-specification</u>	0x00	1
<u> access-response-list-of-data</u>		
<u> SEQUENCE OF</u>	0x03	1
<u> Response for request number 1</u>		
<u> Tag</u>	0x00	1
<u> Response for request number 2</u>		
<u> Tag</u>	0x00	1
<u> Response for request number 3</u>		
<u> Tag</u>	0x00	1
<u> access-response-specification</u>		
<u> SEQUENCE OF</u>	0x03	1
<u> Result for request number 1</u>		
<u> access-response-action</u>	0x03	1
<u> result</u>	0x00	1
<u> Result for request number 2</u>		
<u> access-response-action</u>	0x03	1
<u> result</u>	0x00	1
<u> Result for request number 3</u>		
<u> access-response-action</u>	0x03	1
<u> result</u>	0x00	1

<u>signature-length</u>	0x40	1
<u>signature-content</u>	0x0199E984CEC75DDCA7F1DDF6E53E2E67352A2 BE38A4B66F8ED596606FAB983FF300CAA76DE88 CED9D563A5C03E8F3A7C000780F3F2061C611E9 AAOB18B460D77	64

ECS12 Set Change of Tenancy date on ESME

Command Message Structure

Name	Encoded Content	Encoded Length
MAC Header (general-ciphering)		
<u>tag</u>	0xDD	1
<u>contents</u>	0x0000000000000000	6
<u>ciphered-service</u>		
<u>length</u>	0x54	1
<u>security header</u>		
<u>security control byte (SC)</u>	0x11	1
<u>invocation counter (IC)</u>	0x00000000	4
Grouping Header (general-signing)		
<u>tag</u>	0xDF	1

<u>transaction-id</u>		
<u>length</u>	0x09	1
<u>value (CRA FLAG)</u>	0x01	1
<u>value (Originator Counter)</u>	0x0000000000000002	8
<u>originator-system-title</u>		
<u>length</u>	0x08	1
<u>value</u>	0x123456789ABCDEFO	8
<u>recipient-system-title</u>		
<u>length</u>	0x08	1
<u>value</u>	0xFFFFFFFFFFFFFFFE	8
<u>date-time</u>		
<u>length</u>	0x00	1
<u>other-information</u>		
<u>Length</u>	0x02	1
<u>Message Code</u>	0x0022	2
<u>content</u>		
<u>length</u>	0x20	1
<u>access-request</u>		
<u>tag</u>	0xD9	1
<u>long-invoke-id-and-priority</u>		
<u>configuration</u>	0x20	1
<u>invoke-id</u>	0x000002	3
<u>date-time</u>	0x00	1
<u>access-request-body</u>		

<u>access-request-specification</u>		
<u>SEQUENCE OF</u>	<i>0x01</i>	<i>1</i>
<u>Request number 1</u>		
<u>access-request-set</u>	<i>0x02</i>	<i>1</i>
<u>cosem-attribute-descriptor</u>		
<u> class-id</u>	<i>0x0001</i>	<i>2</i>
<u> instance-id</u>	<i>0x00005E2C0302</i>	<i>6</i>
<u> attribute-id</u>	<i>0x02</i>	<i>1</i>
<u>access-request-list-of-data</u>		
<u>SEQUENCE OF</u>	<i>0x01</i>	<i>1</i>
<u> Parameter for request number 1</u>		
<u> Names</u>		
<u> Tag</u>	<i>0x09</i>	<i>1</i>
<u> Length</u>	<i>0x0C</i>	<i>1</i>
<u> Value</u>	<i>0x07DF0105FF000000008000FF</i>	<i>12</i>
<u>signature-length</u>	<i>0x00</i>	<i>1</i>
<u>mac-content</u>	<i>0xD748D3F87C9764E42D681C11</i>	<i>12</i>

Response Message Structure

Name	Encoded Content	Encoded Length

MAC Header (general-ciphering)		
__tag	0xDD	1
__contents	0x0000000000000000	6
__ciphered-service		
__length	0x40	1
__security header		
__security control byte (SC)	0x11	1
__invocation counter (IC)	0x00000000	4
Grouping Header (general-signing)		
__tag	0xDF	1
__transaction-id		
__length	0x09	1
__value (CRA FLAG)	0x02	1
__value (Originator Counter)	0x0000000000000002	8
__originator-system-title		
__length	0x08	1
__value	0xFFFFFFFFFFFFFFFE	8
__recipient-system-title		
__length	0x08	1
__value	0x123456789ABCDEF0	8
__date-time		
__length	0x00	1
__other-information		

Length	0x02	1
Message Code	0x0022	2
content		
length	0x0C	1
access-response		
tag	0xDA	1
long-invoke-id-and-priority		
configuration	0x20	1
invoke-id	0x0000002	3
date-time	0x00	1
access-request-specification	0x00	1
access-response-list-of-data		
SEQUENCE OF	0x01	1
Response for request number 1		
Tag	0x00	1
access-response-specification		
SEQUENCE OF	0x01	1
Result for request number 1		
access-response-set	0x02	1
result	0x00	1
signature-length	0x00	1
mac-content	0xDF27D0FE42DDED6DC5DCF3F6	12

5663 19 Use Cases

5664 The Use Cases are contained in the embedded HTML document at Table 19.3. Each Use
 5665 Case represents one or more interactions with a Device that make up a GBCS Command,
 5666 Response and / or Alert. This Section 19 provides an overview of the repeatable content
 5667 within these Use Cases.

5668 19.1 Use Case Title

5669 Each Use Case Title section in Table 19.3 provides common information regarding the Use
 5670 Cases that follow. Each section and its purpose is outlined in Table 19.1.

Section	Content
Description	A textual summary of the purpose and scope of the Use Cases encompassed by the Use Case Title
Use Case	Details the Unique Use Case reference, the Use Case name and the Use Case Message Code (see Section 15)
Use Case Cross References	See Section 19.1.1
Use Case Access Permissions	A summary of User Roles that can perform the Use Case. See Section 17 for Remote Party Usage Rights and Section 4.3.2.6 for Trust Anchor Cells applicable. Note that Use Cases from Unknown Remote Parties are performed using the Remote Party Role of Access Control Broker
SMETS / CHTS Objects applicable to Use Case	A list of SMETS / CHTS attributes and associated methods that are applicable to the Use Case. This confirms the properties required by SMETS / CHTS for the attribute/method. This also provides information on the User Gateway Service Request invoked. This table is sorted alphabetically by the entry in the column 'name' concatenated with the entry in the column 'attribute / method'

5671 Table 19.1: SMETS / CHTS content of Use Cases

5672 19.1.1 Use Case Cross Reference Section

5673 Table 19.3 provides an overview of important information relevant to the Use Case. It has a
 5674 structured table as summarised in Table 19.1.1.

Cross Reference	Possible Values	Notes
Remote Party or HAN message	HAN Only Message / Remote Party Message	Needed to identify which GBCS requirements apply. See Section 6
Message Type	Command and Response / Alert with reference to the message categories in Section 6	Needed to identify which GBCS requirements apply
Capable of future dated invocation?	Yes / No	Needed to identify which GBCS requirements apply. See Section 9.2
Requires protection against replay?	Yes / No	Needed to identify which GBCS requirements apply. See Section 4.3.1.5
SEC User Gateway Services	[e.g. 6.20]	Traceability to SEC-listed DCC

Cross Reference	Possible Values	Notes
Schedule (Service Request) Reference	SetDeviceConfiguration(MPxN)]	Service Requests
Read Or Update	Read, Update	Identifies whether the purpose of the Use Case is 'Read' or 'Update'
Response Recipient different from Command Sender	Yes or Blank	Identifies where a Response is sent to a different Remote Party than the originator of the associated Service Request.
Use Case Access Permissions	Supplier (C) Supplier (NC) Supplier prepay top up Network Operator (C) Network Operator (NC) Access Control Broker (NC) WAN Provider (C) Security (C)	Lists which Remote Party Roles may originate the Command within the Use Case. This separates (C) critical and (NC) non critical See Section 17 for more details

5675 Table 19.1.1: Allowable values for SMETS / CHTS Use Case Cross References

5676 19.1.2 Objects Applicable to Use Case Section

5677 This section in Table 19.3 contains a 'SMETS Objects applicable to Use Case' table to
 5678 provide traceability between SMETS functions and methods and the Use Case.

5679 The table contains the values set out in Table 19.1.2.

Row Name	Meaning
Mapping Table row #	Identifier of the SMETS / CHTS object's row in the Mapping Table
Ref	SMETS/CHTS document location of the Attribute (prefixed by the document)
Name	The attribute name as specified in SMETS or CHTS
Attribute / Method	The attribute or method being applied to the SMETS/CHTS
Notes	Describes the Method being applied to the SMETS/CHTS attribute or method in the Use Case.
Sub Category	Specifies whether an attribute or method
Data Type	Details of the data type for the attribute as specified in SMETS or CHTS.

5680 Table 19.1.2: SMETS objects applicable to Use Case

19.1.3 Pre-conditions

5681 Pre-conditions represent conditions for which Device logic is required to ensure correct
 5682 operation of commands contained within a message, on the Device. Exception conditions
 5683 (such as failures) that are managed by the Protocol are not captured as Pre-conditions.
 5684 Manufacturers of Devices must only enforce Pre-conditions that are stated in the Use Cases.
 5685 Note that the use of Pre-conditions is minimised in favour of controls being implemented on
 5686 Service User systems.

5687 19.1.4 Actions

5688 Actions stipulate additional Device actions that must be performed together with successful
 5689 execution of the Use Case.

19.2 Use Case-specific content

Each Use Case is given a unique reference and a title.

19.2.1 DLMS COSEM specific content

Table 19.2.1 sets out the Use Case specific attributes and methods and the DLMS specific mapping.

Within any DLMS COSEM Payload, cosem-attribute-descriptors and cosem-method-descriptors, and associated fields, shall be ordered based on the contents of columns in the Mapping Table. The sort order, described by columns headings in the Mapping Table, shall be:

- first by Update Sequence;
- then by DLMS: Class;
- then by OBIS Code;
- then by Attribute (A) / Method (M); and
- then by Attribute / Method Number.

The sort order shall be ascending in all cases.

For structures, the sequence of elements within a structure shall be as defined in the Blue Book, where it is defined in the Blue Book, or as per the Mapping Table, where it is not defined in the Blue Book.

For clarity, the SMETS / CHTS table in each Use Case is sorted in this same order.

Row Name	Meaning
Mapping table row #	Identifier of the SMETS / CHTS object's row in the Mapping Table
SMETS / CHTS Ref	The section(s) with SMETS/CHTS that refer to the SMETS / CHTS name
SMETS / CHTS Name	The attribute name as specified in SMETS / CHTS
Class	Denotes the DLMS Interface Class
OBIS Code	defines identification codes for all data in DLMS / COSEM compliant metering equipment
Attribute or Method	Denotes whether the row relates to an (A)tttribute or (M)ethod
Attribute / Method Number	Forms part of the attribute identity.
Attribute / Method Name and Blue Book reference	The name given to the DLMS object
DLMS COSEM Data type	The data type assigned to the DLMS object
Constant Value	Where this field is present, this is a fixed value for the life of the Device
Notes	Additional useful information

Table 19.2.1: DLMS mapping for Use Case specific attributes / methods

19.2.2 ZSE specific content

Table 19.3 provides information on the ZSE commands required successfully to complete the Use Case. These must be processed in the order listed in Table 19.2.2.

Table 19.2.2 is grouped by ZSE command.

Row Name	Meaning
Mapping table row #	Identifier of the row in the Mapping Table
SMETS / CHTS Ref	Identifies the SMETS / CHTS section that describes the attribute.
SMETS / CHTS Name	The attribute name as specified in SMETS / CHTS The method being applied to the SMETS / CHTS attribute
Data Type	Identifies the ZSE data type for the attribute
Range	The allowable value range for the attribute
Attribute / Value / Parameter	For ZSE read operations – the attribute or a value returned For ZSE update operations – the attribute or parameter updated.
Cluster :ID	Identifies the ZSE cluster that supports the required functionality
Command :ID	Identifies the command and its unique identifier within the ZSE cluster that is used to read or manipulate the attribute for the purpose of the Use Case. Its ZSE identifier is included
Response :ID	Where specified, this identifies the Response and its unique identifier to the read or update command

Table 19.2.2: ZSE specific content

5713

19.3 Embedded Use Cases

5714 Table 19.3 contains the Use Cases that fulfil the interface requirements to cover Commands
 5715 (and their Responses) and Alerts (where applicable). In addition, it includes ZSE Message
 5716 Templates.

5717 Note: DLMS COSEM methods that have values which have an impact on the execution of
 5718 the method (that is, methods with input values that are not integer(0)), the DLMS part of the
 5719 Mapping Table and the Use Case include two or more rows. One row contains the method,
 5720 and the subsequent row(s) contain the value(s) to be sent with the method.

5721 A number of Use Cases are also covered in GBCS main body. These are identifiable from
 5722 the Table of Contents.



GBCS v0.8.1

SMETS2_UseCases.ht

5723

Table 19.3: Use Cases

5725 20 Mapping Table

5726 Table 20 contains the Mapping Table from which the Use Cases and Message Templates
5727 were generated. These tables map between SMETS attributes and methods, SEC Service
5728 Requests, Use Cases, DLMS COSEM attributes and methods and ZSE clusters, attributes
5729 and commands.

5730 In addition to the Use Cases, certain columns in the Mapping Table are directly referenced
5731 from this document.

5732 Please note that only rows marked 'E' (External to HAN) in column F are fully specified,
5733 since those rows relate to Remote Party Messages. Other rows are only specified to the
5734 extent that these elements of Remote Party Messages rely on them.



GBCS v0.8.1 SMETS 2
5735 requirements mapping

5736 Table 20: Mapping Table

21 Glossary

||

X || Y shall mean the concatenation of the two octet strings X and Y.

Abstract Syntax Notation One (ASN.1)

ASN.1 is a standard notation for the definition of data types and values. A data type (or type for short) is a category of information (for example, numeric, textual, still image or video information). A data value (or value for short) is an instance of such a type. ASN.1 defines several basic types and their corresponding values, and rules for combining them into more complex types and values. In some protocol architectures, each message is specified as the binary value of a sequence of octets. However, standards-writers need to define quite complex data types to carry their messages, without concern for their binary representation. In order to specify these data types, they require a notation that does not necessarily determine the representation of each value. ASN.1 is such a notation.

Access Control Broker (ACB)

In the context of a specific Device, the Known Remote Party whose Security Credentials are stored in the {accessControlBroker, digitalSignature, management} Trust Anchor Cell where present, and stored in the {accessControlBroker, keyAgreement, management} Trust Anchor Cell otherwise.

The ACB applies Cryptographic Protections to all Commands addressed to the Device in question, except potentially for certain recovery scenarios catered for by the Security Credentials Commands.

Access Control Broker to Device MAC (ACB-SMD MAC)

A MAC generated by the Access Control Broker in relation to a Command which can only be verified by the Device which is the target of the Command.

Activate Emergency Credit

Command described in SMETS.

Additional Authenticated Data (AAD)

One of the inputs to the calculation of a MAC. The AAD is protected by the MAC but is not encrypted. AAD has the same meaning as in *NIST Special Publication 800-38D*: <http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf>.

Alert

A Message generated by a Device including in response to a problem or the risk of a potential problem.

Alert Code

A 16 bit unsigned integer taking the values specified in Section 16. The Alert Code and Event Code are the same for a given Event.

Application Association

Shall have the meaning specified in the DLMS COSEM standards.

Application Layer Protocol Data Unit (APDU)

Information delivered as a unit among peer entities of networks.

Association LN Object

A DLMS Component specified in the Blue Book which provides role based access control.

- 5779 **Authenticated Decryption**
5780 Has the same meaning as specified in *NIST Special Publication 800-38D*:
5781 <http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf>
- 5782 **Authenticated Encryption (AE)**
5783 Has the same meaning as specified in *NIST Special Publication 800-38D*:
5784 <http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf>
- 5785 **Authentication**
5786 The method used to confirm the identity of entities or Devices wishing to communicate and
5787 'Authenticated' and 'Authenticity' shall be construed accordingly.
- 5788 **Authentication Key**
5789 Shall be as defined in the Green Book.
- 5790 **Authorisation**
5791 The process of granting access to a resource and 'Authorised' shall be construed
5792 accordingly.
- 5793 **Authorised Public Key Infrastructure (APKI)**
5794 A key infrastructure that is compliant with the Certificate related requirements of this GBCS.
- 5795 **Auxiliary Load Control Switch**
5796 A switch or other means of controlling a load on the Supply.
- 5797 **Boost Function**
5798 ESME functionality described in SMETS.
- 5799 **Business Originator**
5800 The Smart Metering Entity sending the first Message in a Use Case.
- 5801 **Business Target**
5802 The Smart Metering Entity receiving the first Message in a Use Case.
- 5803 **Certificate**
5804 An electronic document that binds an identity, and possibly other information, to a Public Key.
- 5805 **Certification Request**
5806 A message requesting the issue of a Certificate by a Certification Authority.
- 5807 **Certification Authority (CA)**
5808 A trusted entity which issues Certificates.
- 5809 **Certification Authority Certificate**
5810 A Certificate issued to a Certification Authority that allows Certification Path Validation in
5811 relation to Remote Party's Certificates.
- 5812 **Certification Path Validation**
5813 Shall have the meaning defined in Section 4.3.2.8.
- 5814 **Certification Revocation List (CRL) Validation**
5815 Shall have the meaning defined in Section 4.3.2.8 and shall be undertaken by the Access
5816 Control Broker and not Devices.
- 5817 **Ciphered Information**

5818 Shall have the meaning defined in Section 8.4.

5819 **Ciphertext**

5820 An output of the Authenticated Encryption function and an input of the Authenticated
5821 Decryption function defined in *NIST Special Publication 800-38D*. The unencrypted form of
5822 the Ciphertext is the Plaintext.

5823 **Clock**

5824 A timing mechanism which has a minimum resolution of 1 second.

5825 **Command**

5826 An instruction to perform a function received or sent by any interface.

5827 **Command Response Alert (CRA) Flag**

5828 An element within a Message Header that enumerates whether the Message is a Command
5829 or a Response or an Alert

5830 **Communications Hub**

5831 A Device complying with the CHTS.

5832 **Communications Hub Function (CHF)**

5833 The functionality in the Communications Hub specific to its operation as a bridge between
5834 the WAN Interface and the HAN Interface.

5835 **Communications Hub Technical Specifications (CHTS)**

5836 The document designated by the Secretary of State to describe the minimum capabilities of
5837 Communications Hubs installed to satisfy the roll-out licence conditions.

5838 **Confidentiality**

5839 The state of information, in transit or at rest, where there is assurance that it is not
5840 accessible by Unauthorised parties through either unintentional means or otherwise.

5841 **Consumer**

5842 A person who lawfully resides at the Premises that is being Supplied.

5843 **Consumer Access Device (CAD)**

5844 A Device which, in terms of this GBCS, supports the same Messages as an IHD.

5845 **Consumption**

5846 In the context of GSME, Gas Consumption or in the context of ESME, Electricity
5847 Consumption information.

5848 **Contingency Key**

5849 A feature of Trust Anchor Management Protocol (RFC 5934), and only ever used in a
5850 recovery scenario when the root Certificate (Apex Trust Anchor) needs to be replaced.

5851 **Critical Message**

5852 A Remote Party Message which may relate to supply being affected, financial fraud or the
5853 compromise of Device security. Critical, Critical Commands, Critical Alerts and Critical
5854 Responses shall be construed accordingly.

5855 **Cryptographic Algorithm**

5856 An algorithm for performing one or more cryptographic functions which may include
5857 Encryption; Decryption; digitally signing or Hashing of information, data, or messages; or
5858 exchange of Security Credentials.

5859 **Cryptographic Protection**

5860 A part of a Message constructed to provide assurance to the Message recipient in terms of
5861 one or more of integrity, authenticity, non-repudiation and Confidentiality.

5862 **Currency Units**

5863 The units of monetary value in major and minor units.

5864 **Current Private Key**

5865 A Device Private Key for which the Device has successfully received and processed a
5866 Certificate for the corresponding Public Key as defined in Section 13.5.

5867 **Data and Communications Company (DCC)**

5868 The holder of the licence for the provision of a smart meter communication service granted
5869 pursuant to section 6(1)(f) or 6(1A) of the Electricity Act 1989 or section 7AB of the Gas Act
5870 1986.

5871 **Data Store**

5872 An area of a Device capable of storing information for future retrieval.

5873 **Decryption**

5874 The process of converting Encrypted information by an Authorised party to recover the
5875 original information. Like terms shall be construed accordingly.

5876 **Device**

5877 A Device that is one of ESME, GSME, Gas Proxy Function, Communications Hub Function,
5878 Type 1 Device or a Type 2 Device.

5879 **Device Based Access Control (DBAC)**

5880 Shall have the meaning defined in Section 13.7.3.

5881 **Device Certificate**

5882 Shall have the meaning set out in Section 12.

5883 **Device Certification Authority (DCA)**

5884 A trusted entity which issues Device Certificates.

5885 **Device Log**

5886 Data item described in SMETS and CHTS.

5887 **Device Specifications**

5888 The document set comprising SMETS (including the IHDTs, HCALCSTS and PPMIDTS),
5889 and CHTS.

5890 **Digital Signature**

5891 The information appended to a Message which is created using the sender's Private Key,
5892 can be verified using the corresponding Public Key contained in the sender's Certificate, and
5893 provides the receiver with assurance that the sender is who they claim to be, the message
5894 has not been altered in transit and that the sender sent the Message.

5895 **Digital Signing**

5896 The creation of a Digital Signature.

5897 **Digital Signing Certificate**

5898 A Certificate which states that the Public Key contained within, and its associated Private
5899 Key, may be used for Digital Signing purposes.

5900 **Distinguished Encoding Rules**

5901 Shall have the meaning defined in <http://www.itu.int/ITU-T/studygroups/com17/languages/X.690-0207.pdf>

5903 **DLMS COSEM**

5904 Device Language Message Specification / Companion Specification for Energy Metering - an
5905 Application Layer protocol.

5906 **Elliptic Curve DSA (ECDSA)**

5907 The Elliptic Curve Digital Signature Algorithm forming part of the NSA Suite B standard (see
5908 (<http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf>) as specified in Section 4.3.3

5909 **Encoding(X)**

5910 The encoding of a variable length integer X, as specified in Section 3.3.

5911 **Encryption**

5912 The process of converting information in order to make it unintelligible other than to
5913 Authorised parties. Like terms shall be construed accordingly.

5914 **Encryption Originator Counter**

5915 Shall have the meaning defined in Table 23.

5916 **Encryption Remote Party**

5917 The Remote Party that encrypted Encrypted data items.

5918 **Entity Identifier**

5919 A 64 bit unsigned integer uniquely identifying a Smart Metering Entity.

5920 **ESME**

5921 Electricity Smart Metering Equipment, as described in the SMETS.

5922 **Event**

5923 A change in state generated by a Device in response to an internal or external trigger.

5924 **Event Code**

5925 A 16 bit unsigned integer taking the values specified in Section 16. The Alert Code and
5926 Event Code are the same for a given Event.

5927 **Event Log**

5928 Data item described in SMETS and CHTS.

5929 **Execution Counter**

5930 Shall have the meaning defined in Section 4.3.1.5.

5931 **Firmware**

5932 The embedded software programmes and / or data structures that control Devices.

5933 **Force Replace**

5934 The means to instruct a Communications Hub to replace an ESME or GSME Firmware
5935 image that it holds, e.g. when the image has only been partially downloaded to the ESME or
5936 GSME. This enables recovery from failures.

5937 **Gas Proxy Function (GPF)**

5938 The Gas Proxy Function as defined in the Communications Hub Technical Specification.

5939 **Galois Counter Mode (GCM)**

5940 The mode of operation specified in *NIST Special Publication 800-38D*.

5941 **GBZ**

5942 A set of structures in the GBCS which carry ZCL / ZSE commands.

5943 **General Block Transfer (GBT) / GBT Message**

5944 General Block Transfer is a DLMS COSEM mechanism for decomposing APDUs above
5945 maximum sizes that can be transported in to a number of smaller APDUs, which are no
5946 larger than the maximum sizes. A GBT Message is one of these smaller APDUs

5947 **GMAC**

5948 Variant of GCM that is used to generate Message Authentication Code from non-
5949 Confidential data, as specified in *NIST Special Publication 800-38D*.

5950 **GSME**

5951 Gas Smart Metering Equipment, as described in the SMETS.

5952 **HAN Only Message**

5953 A Message where both the sender and recipient are Devices on the same Smart Metering
5954 Home Area Network.

5955 **HAN Connected Auxiliary Load Control Switch (HCALCS)**

5956 A Type 1 Device capable of communicating with an ESME.

5957 **Hashing**

5958 A repeatable process to create a fixed size condensed representation of a Message or any
5959 arbitrary data, as further set out in Section 4.3.3. Hash and like terms shall be construed
5960 accordingly.

5961 **HCALCS**

5962 HAN Connected ALCS.

5963 **HCALCSTS**

5964 The HAN Connected Auxiliary Load Control Switches (HCALCS) Technical Specification.

5965 **Highest Prior Sequence Number**

5966 Shall have the meaning defined in Section 13.3.5.3.

5967 **Home Area Network Interface (HAN Interface)**

5968 A component of GSME, ESME, IHD or other Consumer Device that is capable of sending
5969 and receiving information to and from other Devices.IHDTs

5970 The In Home Display Technical Specifications.

5971 **IHD**

5972 In Home Display.

5973 **IHD Source Device**

5974 An ESME or GPF.

5975 **Initialization Vector (IV)**

5976 An input to the Authenticated Encryption and Authenticated Decryption functions defined in
5977 *NIST Special Publication 800-38D*. Where the GBCS applies, it shall have the values as
5978 specified at Section 4.3.3.4.

5979 **Inter-PAN**

5980 Glossary term defined in CHTS.

5981 **Join**

5982 The process of authorising two Devices to communicate at the application layer.

5983 **Key**

5984 Data used to determine the output of a cryptographic operation.

5985 **Key Agreement**

5986 A means to calculate a shared Key between the two parties.

5987 **Key Agreement Certificate**

5988 A Certificate which states that the Public Key contained within, and its associated Private
5989 Key, may be used for Key Agreement purposes.

5990 **Key Derivation Function (KDF)**

5991 A function to generate derived keying material from a Shared Secret and other information

5992 **Known Remote Party (KRP)**

5993 In the context of a specific Device, a Remote Party whose Security Credentials are stored on
5994 that Device in at least one Trust Anchor Cell.

5995 **KRP Signature**

5996 A Digital Signature generated by a Known Remote Party.

5997 **Len(X)**

5998 The number of octets in the variable length octet string X.

5999 **MAC Header**

6000 As defined in Section 6, a part of a message which is only present when the Message
6001 contains a MAC but which is additional to the MAC.

6002 **Manufacturer Hash Image**

6003 Shall have the meaning defined in Section 11.2.4.

6004 **Mapping Table**

6005 The spreadsheet detailing Use Cases and associated protocol requirements as embedded in
6006 Section 20.

6007 **Maximum Credit Threshold**

6008 Shall have the meaning defined in SMETS.

6009 **Maximum Meter Balance Threshold**

6010 Shall have the meaning defined in SMETS.

6011 **Message**

6012 A Command, Response or Alert.

6013 **Message Authentication**

6014 The process by which the receiver of a Message is provided with assurance that the sender
6015 is who they claim to be and that the Message is in the form originally sent.

6016 **Message Authentication Code (MAC)**

6017 The number incorporated in a Message to provide Message Authentication, as set out in
6018 Section 4.3.3.

6019 **Message Category**

6020 A grouping of Remote Party Messages.

6021 **Message Code**

6022 A 16 bit unsigned integer identifying the Use Case that the Message in question must
6023 conform to. Message Codes have the values specified in Section 15.

6024 **Message Identifier**

6025 Message Identifier shall be the concatenation of:

- 6026 • Business Originator ID;
- 6027 • Business Target ID;
- 6028 • CRA Flag; and
- 6029 • Originator Counter.

6030 **Message Series**

6031 Shall have the meaning defined in Section 7.2.11.1.

6032 **Message Template**

6033 A protocol-specific table defining the encoding of a Message.

6034 **Message Type**

6035 The Message Types are Command, Response or Alert.

6036 **Network Interface**

6037 A WAN Interface or HAN Interface.

6038 **Network Operator**

6039 In the context a specific Device, the Known Remote Party whose Security Credentials are
6040 stored in the {networkOperator, digitalSignature, management} Trust Anchor
6041 Cell.

6042 **Object Identifier (OID)**

6043 An identifier used to name an object. Structurally, an OID consists of a node in a
6044 hierarchically-assigned namespace, formally defined using the ASN.1 standard.

6045 **Organisation Certificate**

6046 Shall have the meaning set out in Section 12.

6047 **Originator Counter**

6048 Shall have the meaning defined in Section 4.3.1.2.

6049 **OtherInfo**

6050 An input to the KDF with the meaning as specified in Section 5.8.1 of *NIST Special*
6051 *Publication 800-56Ar2*.

- 6052 **Other User**
6053 A Remote Party which is not a Known Remote Party in relation to any Device, and so is
6054 always an Unknown Remote Party in any communication with a Device.
- 6055 **Outcome**
6056 The result of executing a Command, expressed as success or failure.
- 6057 **Payload**
6058 Part of the Message that provides the message-specific content.
- 6059 **Payment Mode**
6060 The information held on GSME as described at section 4 in the Smart Metering Equipment
6061 Technical Specifications.
- 6062 **Pending Private Key**
6063 A Device Private Key for which the Device has not successfully received and processed a
6064 Certificate for the corresponding Public Key as defined in Section 13.5.
- 6065 **Personal Data**
6066 Any information comprising Personal Data as such term is defined in the Data Protection Act
6067 1998 at the date the GBCS is brought into force.
- 6068 **Pre-Payment Interface Device (PPMID)**
6069 A Device that provides a User Interface for Prepayment Mode related information and
6070 Commands.
- 6071 **Plain Format**
6072 A Signature is a pair of integers, r and s. For the Elliptic Curve required by the GBCS, each
6073 can be represented as a 256 bit (or 32 octet) string. The Plain Format of a GBCS signature
6074 is the concatenation R || S where R is the 32 octet string representing r and S is the 32 octet
6075 string representing s. Thus, a GBCS Signature is an octet string of length 64.
- 6076 **Plaintext**
6077 An input to the Authenticated Encryption function and an output from the Authenticated
6078 Decryption function defined in *NIST Special Publication 800-38D*. Plaintext is the data
6079 whose Confidentiality is to be protected by Encryption. The encrypted form of the Plaintext is
6080 the Ciphertext.
- 6081 **PPMIDTS**
6082 The Prepayment Interface Device (PPMID) Technical Specification.
- 6083 **Polyphase Electricity Metering Equipment**
6084 Electricity metering equipment containing three measuring elements suitable for a polyphase
6085 supply with up to three phases and neutral.
- 6086 **Premise(s)**
6087 The premise(s) which is / are being Supplied.
- 6088 **Prepayment Daily Read Log**
6089 Data item described in SMETS.
- 6090 **Prepayment Token Decimal (PPTD)**
6091 Shall have the meaning defined in Section 14.1.
- 6092 **Prepayment Top-Up Token**

6093 Shall have the meaning defined in Section 14.1.

6094 **Private Digital Signing Key**

6095 A Private Key used for Digital Signing only.

6096 **Private Key**

6097 The key in a Public-Private Key Pair which must be kept secure by the entity to which it
6098 relates.

6099 **Private Key Cell**

6100 Shall have the meaning defined in Section 4.3.2.3. A Private Key Cell may be Current or
6101 Pending.

6102 **Private Key Agreement Key**

6103 A Private Key used for Key Agreement only.

6104 **Protection Against Replay**

6105 An attribute defined in a Use Case specifying whether a recipient Device is required to
6106 implement the Protection Against Replay mechanisms, as defined in Section 4.3.1.5, for the
6107 Command covered by the Use Case.

6108 **Protocol Data Unit (PDU)**

6109 Information delivered as a unit among peer entities of networks containing control information,
6110 address information or data.

6111 **Public Digital Signing Key**

6112 A Public Key used for Digital Signing only.

6113 **Public Key**

6114 The key in a Public-Private Key Pair which can be distributed to other parties.

6115 **Public Key Agreement Key**

6116 A Public Key used for Key Agreement only.

6117 **Public Key Security Credentials**

6118 Security Credentials which include a Public Key.

6119 **Public-Private Key Pairs**

6120 Two mathematically related numbers that are used in Cryptographic Algorithms.

6121 **Recovery**

6122 In the context a specific Device, the Known Remote Party whose Security Credentials are
6123 stored in the {recovery, digitalSignature, management} Trust Anchor Cell.

6124 **Reliable Time**

6125 The state of the Device clock such that is within 10 seconds of UTC, synchronised with the
6126 HAN time server and confirmed by Set Clock Command from the Remote Party whose
6127 security Credentials are stored in the {supplier, digitalSignature, management}
6128 Trust Anchor Cell

6129 **Remote Party**

6130 An entity which is remote from the Premises and is able to either send Messages to or
6131 receive Messages from a Device within the Premises, whether directly or via a third party.

6132 **Remote Party Alert**

- 6133 Shall have the meaning defined in Section 7.2.3.
- 6134 **Remote Party Command**
- 6135 Shall have the meaning defined in Section 7.2.1.
- 6136 **Remote Party Message**
- 6137 A Message where either the sender(s) or recipient(s) are not Devices.
- 6138 **Remote Party Role**
- 6139 A class of Remote Party in relation to which one or more Devices is capable of storing
- 6140 Security Credentials.
- 6141 **Remote Party Role Code**
- 6142 An 8 bit unsigned integer which uniquely identifies a Remote Party Role. The value for each
- 6143 Remote Party Role shall be as defined in Section 4.3.2.4.
- 6144 **Replay Attack**
- 6145 A form of attack on a Communications Link in which a valid information transmission is
- 6146 repeated through interception and retransmission.
- 6147 **Response**
- 6148 Sent on, or received from the User Interface or HAN Interface or any other interface
- 6149 containing information in response to a Command.
- 6150 **Response Payload**
- 6151 The parts of a Response that are not related to Cryptographic Protections for integrity,
- 6152 authenticity or non-repudiation, as defined in Section 7.2.2.
- 6153 **Role**
- 6154 The entitlement of a party to execute one or more Commands.
- 6155 **Root**
- 6156 In the context a specific Device, the entity whose Security Credentials are stored in the
- 6157 {root, keyCertSign, management} Trust Anchor Cell.
- 6158 **Secure Perimeter**
- 6159 **A physical border surrounding ESME, GSME or the PPMID.Security Credential Document**
- 6160 A Security Credential Document shall be defined as either a:
- 6161 • Device's Certificate; or a
- 6162 • Remote Party's Certificate; or a
- 6163 • Certification Authority Certificate
- 6164 **Security Credentials**
- 6165 Information used to identify and / or Authenticate a Device, Party or system.
- 6166 **Security Log**
- 6167 Data item described in SMETS and CHTS.
- 6168 **SHA-256**
- 6169 The Hashing algorithm of that name approved by the NIST (see
- 6170 http://csrc.nist.gov/groups/ST/toolkit/secure_hashing.html).
- 6171 **Shared Secret**

- 6172 A number which is established by two parties through the Key Agreement technique
6173 specified in this GBCS and which can be used as input to a KDF.
- 6174 **Shared Secret Key**
- 6175 **A number which is derived using the KDF specified in this GBCS. Smart Metering Device to**
6176 **Known Remote Party MAC (SMD-KRP MAC)**
- 6177 A MAC generated by a Device in relation to a Response or Alert which can only be verified
6178 by the Known Remote Party which is the target of the Response or Alert.
- 6179 **Smart Metering Entity**
- 6180 An entity that is either a Device or a Remote Party.
- 6181 **Smart Metering Equipment Technical Specifications (SMETS)**
- 6182 The document designated by the Secretary of State to describe the minimum capabilities of
6183 equipment installed to satisfy the roll-out licence conditions.
- 6184 **Smart Metering Home Area Network (SMHAN)**
- 6185 The network enabling communications between the Devices recorded within a
6186 Communications Hubs' Device Log (as defined in CHTS).
- 6187 **SMD Signature**
- 6188 A Digital Signature generated by a Device.
- 6189 **Supplementary Originator Counter**
- 6190 Shall have the meaning defined in Section 23.
- 6191 **Supply**
- 6192 The supply of gas to Premises for GSME and the supply of electricity to Premises for ESME
6193 and 'Supplied' shall be construed accordingly.
- 6194 **Supplier**
- 6195 A person authorised by licence to Supply gas to Premises for GSME and a person
6196 authorised by licence to Supply electricity to Premises for ESME. In the context of a specific
6197 Device, the Known Remote Party whose Security Credentials are stored in the {supplier,
6198 digitalSignature, management} Trust Anchor Cell.
- 6199 **Tag**
- 6200 The first element within a Message Header or part of a Message that provides identification
6201 of the Message or part of Message that follows.
- 6202 **Tariff**
- 6203 The structure of prices and other charges relating to a Supply.
- 6204 **Tariff Block Counter Matrix**
- 6205 Data item described in SMETS.
- 6206 **TOU**
- 6207 Time of Use.
- 6208 **Transactional Atomicity**
- 6209 The type and order of the constituent parts of a Command.
- 6210 **Transitional Change of Supplier**

6211 In the context a specific Device, the Known Remote Party whose Security Credentials are
6212 stored in relation to the Transitional Change of Supplier role.

6213 **Trust Anchor (TA)**

6214 A Trust Anchor represents a Remote Party via a Public Key and associated data stored on a
6215 Device. A Trust Anchor is used by the Device in specified cryptographic operations to
6216 determine whether it should act on Remote Party Commands received.

6217 **Trust Anchor Cell**

6218 A data store on a Device capable of storing one Trust Anchor. Each Trust Anchor Cell is for
6219 a fixed and pre-specified KeyUsage, CellUsage and RemotePartyRole.

6220 **Trust Anchor Management (TAMP)**

6221 A range of IETF RFCs relate to Trust Anchor Management, including:

- 6222 • [RFC4210] Adams, C., Farrell, S., Kause, T., and T. Mononen, 'Internet X.509 Public
6223 Key Infrastructure Certificate Management Protocol (CMP)', [RFC 4210](#), September
6224 2005.
- 6225 • [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk,
6226 'Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL)
6227 Profile', [RFC 5280](#), May 2008.
- 6228 • [RFC5914] Housley, R., Ashmore, S., and C. Wallace, 'Trust Anchor Format', [RFC
6229 5914](#), June 2010.
- 6230 • [RFC5934] Housley, R., Ashmore, S., and C. Wallace, 'Trust Anchor Management
6231 Protocol (TAMP)', [RFC 5934](#), August 2010.
- 6232 • [RFC6024] Reddy, R. and C. Wallace, 'Trust Anchor Management Requirements', [RFC
6233 6024](#), October 2010.

6234 **Trusted Source**

6235 A source whose identity is confidently and reliably validated.

6236 **Twin Element Electricity Metering Equipment**

6237 Electricity metering equipment containing two measuring elements.

6238 **Type 1 Device**

6239 A Device, other than GSME, ESME, Communications Hub, Communications Hub Function
6240 or Gas Proxy Function, that stores and uses the Security Credentials of other Devices for the
6241 purposes of communicating with them via its HAN Interface.

6242 **Type 2 Device**

6243 A Device that does not store or use the Security Credentials of other Devices for the
6244 purposes of communicating with them via its HAN Interface.

6245 **Unauthorised**

6246 Not Authorised.

6247 **Unauthorised Physical Access**

6248 Unauthorised access to the internal components of any Device within GSME or ESME
6249 through its Secure Perimeter.

6250 **Unique Transaction Reference Number (UTRN)**

6251 A 20 decimal digit number that is used to convey a Pre-Payment Top-Up Remote Party
6252 Command to an ESME / GSME.

- 6253 **Unknown Remote Party (URP)**
6254 In the context of a specific Device, a Remote Party whose Security Credentials are not
6255 stored on that Device.
- 6256 **Upgrade Image**
6257 Shall have the meaning defined in Section 11.2.2.
- 6258 **Use Case**
6259 The structure, format and processing of a Message.
- 6260 **User Interface**
6261 An interface for providing local human interaction with Devices which supports input and
6262 visual output.
- 6263 **User Interface Command**
6264 A Remote Party Command that is entered through the User Interface.
- 6265 **UTC**
6266 Coordinated Universal Time.
- 6267 **UTRN Check Digit**
6268 Shall have the meaning defined in Section 14.1.
- 6269 **UTRN Counter Cache**
6270 Shall have the meaning defined in Section 14.1.
- 6271 **Variant Message**
6272 A Message that does not fall in to any of the Message Categories defined in Section 6.
- 6273 **Wide Area Network (WAN) Interface**
6274 A component of a Communications Hub that is capable of sending and receiving information
6275 via the Wide Area Network Provider.
- 6276 **Wide Area Network (WAN) Provider**
6277 The organisation providing communications over the WAN Interface of the Communications
6278 Hub. Consequently, in the context of a specific Communications Hub, the Known Remote
6279 Party whose Security Credentials are stored in the {wanProvider, digitalSignature,
6280 management} Trust Anchor Cell.
- 6281 **ZigBee Cluster Library (ZCL)**
6282 The ZigBee Cluster Library Specification reference document as defined in the
6283 'Documentation Alignment' section of this GBCS.
- 6284 **ZigBee SE (ZSE)**
6285 The ZigBee Smart Energy Profile Specification as defined in the 'Documentation Alignment'
6286 section of this GBCS.

22 Annex 1 – Additional DLMS Class

6287 The class described below shall be supported by ESME. Extended Data (class_id: 9000
 6288 version: 0)

Attribute(s)			Data type	Min.	Max.	Def.
1.	logical_name	(static)	octet-string[6]			
2.	value_active	(dyn.)	CHOICE			
3.	scaler_unit_active	(dyn.)	scal_unit_type			
4.	value_passive	(static)	CHOICE			
5.	scaler_unit_passive	(static)	scal_unit_type			
6.	activate_passive_value_time	(static)	octet-string			
Methods(s)		Data type				
1.	reset(data)		Integer			
2.	activate_passive_value(data)		integer			

6290 22.1 Attribute description

logical_name	Identifies the 'Data' object instance
	Contains the data.
	<i>CHOICE</i>
	{
	-- simple data types
	null-data [0],
	Boolean [3],
	bit-string [4],
	double-long [5],
	double-long-unsigned [6],
	octet-string [9],
	visible-string [10],
	UTF8-string [12],
	Bcd [13],
	integer [15],
	long [16],
	unsigned [17],
	long-unsigned [18],
	long64 [20],
	long64-unsigned [21],
	enum [22],
	float32 [23],
	float64 [24],
	date-time [25],
	date [26],
	time [27],
	-- complex data types
	array [1],
	structure [2],
	compact-array [19]
	}
value_active	Provides information on the unit and the scalar for the value. scal_unit_type: structure

	<pre>{ scalar, unit } scalar: integer This is the exponent (to the base of 10) of the multiplication factor unit: enum Enumeration defining the physical unit; for more information check the Blue Book</pre>
	<p>Contains the data.</p> <p><i>CHOICE</i></p> <pre>{ -- simple data types null-data [0], Boolean [3], bit-string [4], double-long [5], double-long-unsigned [6], octet-string [9], visible-string [10], UTF8-string [12], Bcd [13], integer [15], long [16], unsigned [17], long-unsigned [18], long64 [20], long64-unsigned [21], enum [22], float32 [23], float64 [24], date-time [25], date [26], time [27], -- complex data types array [1], structure [2], compact-array [19]</pre>
value_passive	<p>}</p> <p>The data type depends on the instantiation defined by the 'logical name'. It has to be chosen so, that together with the logical name, an unambiguous interpretation is possible.</p>
	<p>Provides information on the unit and the scalar for the value.</p> <p><i>scal_unit_type</i>: structure</p> <pre>{ scalar, unit } scalar: integer This is the exponent (to the base of 10) of the multiplication factor unit: enum Enumeration defining the physical unit; for more information check the Blue Book</pre>
scaler_unit_passive	<p>Defines the time when the object itself calls the specific method <i>activate_passive_value</i>. A definition with 'not specified' notation in all fields of the attribute will deactivate this automatic activation. Partial 'not specified' notation in just some fields of date and time is not allowed.</p> <p>octet-string, formatted as set in 4.1.6.1 for <i>date_time</i> of the Blue DLMS Book</p>
activate_passive_value_time	

22.2 Method description

Reset	This method forces a reset of the object. By invoking this method, the value is set to the default value. The default value is an instance specific constant. data ::= integer(0)
--------------	--

activate_passive_value	This method copies all attributes called ..._passive to the corresponding attributes called ..._active. data ::= integer(0)
-------------------------------	--

6292

6293

23 Annex 2 - Counters and their use in transaction identification and Protection Against Replay protection - informative

6296

6297

6298

Table 23 provides a summary of the Counters used in GB Smart Metering and outlines the purpose each serves in providing transaction identity, traceability and Protection Against Replay protection. These are fully detailed in Section 4.3.1 and Section 14 and are provided here as a review aid only.

Name	Description	Purpose	Impact on Device
[Remote Party] Originator Counter	The KRP or the ACB's Originator Counter. Originator Counters are always strictly numerically greater than any previous Originator Counter from that Message originator to the targeted Device. Originator Counters shall not use the UTRN reserved range unless as part of a Prepayment Top Up Command. Remote Parties may choose to increment a UTRN Originator Counter separately from other Originator Counters.	The Originator Counter provides a unique Message identity (in combination with CRA Flag, sender id and recipient id). The Originator Counter is also used as an input value for symmetric Key Derivation Functions. The Originator Counter is used for Protection Against Replay protection.	The highest accepted value is stored as the Execution Counter or in the UTRN Counter cache as appropriate.
[Device] Originator Counter	A Device's Originator Counter This must be strictly numerically greater than any previous Originator Counter from that Device.	The Originator Counter provides a unique Message Identity (in combination with CRA Flag, sender id and recipient id) The Originator Counter is also used as an input value for symmetric Key Derivation Functions.	The Device shall ensure that the value it generates (e.g. for Alerts) is strictly numerically greater than any previous Originator Counter value or Supplementary Originator Counter value it has placed in any previous Message it has generated.
Supplementary Remote Party Counter	The Originator Counter (or reference) of an Unknown Remote Party requesting the service from the ACB.	The Supplementary Remote Party Counter supports Message identification of Responses by the URP as the originator of the service request associated to the Command.	The Supplementary Remote Party Counter is incorporated into the corresponding Response by the Device. The Response also contains the Originator Counter of the ACB

Supplementary Originator Counter	<p>The Supplementary Originator Counter is a Device generated number which is strictly numerically greater than any previous Supplementary Originator Counter or Originator Counter placed in previous Messages by the Device). This is used in response to Commands as specified in Section 4.3.1.4 (URP accessible Commands where the response contains sensitive values).</p>	<p>The Supplementary Originator Counter is used in a Response to a Command from an URP for the generation of symmetric keys for use in MAC creation and Encryption of sensitive values.</p>	<p>The Device shall ensure that the value it generates (e.g. for Alerts) is strictly numerically greater than any previous Originator Counter or Supplementary Originator Counter values it has used in any previous Message.</p> <p>The Supplementary Originator Counter may be the same as Originator Counter in any given Message but this is an implementation decision).</p>
Execution Counter	<p>The Execution Counter is the last accepted Originator Counter value for commands requiring Protection Against Replay and which cannot be future dated. It is stored by the Device for each Remote Party/Command combination.</p> <p>Note that only the Supplier (or for CHF the WAN Provider) can send Commands that require Protection Against Replay with the exception of the Update Security Credentials Command which can be sent by multiple roles.</p>	<p>The Execution Counter is used to support Protection Against Replay of Commands for immediate execution.</p> <p>Where Commands are protected from Protection Against Replay then Devices will reject Commands where the Originator Counter in the Command not greater than the existing value of the Execution Counter stored on the Device.</p>	<p>Each Device will store an Execution Counter value for each KRP/Command-type combination.</p>
UTRN Counter	<p><u>The UTRN Counter is detailed separately in Section 14 but a summary is included here for completeness.</u></p> <p>The UTRN Counter must be strictly greater by one than the highest previous UTRN Counter issued for the target Device by the KRP</p> <p>The UTRN Counter comprises the 32 most significant bits of the Originator Counter (this is a reserved range of Originator Counters where the least significant 32 bits are set to 0) which is included in a Pre-payment Top-Up Command (whether entered locally or received over the WAN).</p> <p>The Device checks:</p>	<p>The UTRN Counter provides a specific Protection Against Replay mechanism for pre-pay</p> <p>Where the Command is received over the WAN, the Originator Counter (and therefore the UTRN Counter) is as contained in the WAN Prepayment Top Up Command.</p> <p>If the UTRN Counter contained within a prepayment Command (whether entered locally or received over the WAN) is already in the UTRN Counter Cache or is less than the lowest value in the UTRN Counter Cache on the Device, then Devices will reject the UTRN.</p>	<p>Each ESME and GSME must maintain a UTRN Counter Cache as an array of the last 100 UTRN Counter entries. Where the array is full, the numerically lowest value in the array is overwritten.</p>

	<ul style="list-style-type: none"> that the UTRN Counter contained within a UTRN is greater than the lowest value in the UTRN Counter cache held on the Device. This ensures that a limited number of UTRNs can be executed out of sequence); and that the UTRN Counter is not equal to any value currently held in the UTRN Counter cache, i.e. that the Pre-payment Command has not be accepted before. 		
PTUT Truncated Originator Counter	<p><u>The PTUT Truncated Originator Counter is detailed separately in Section 14 but a summary is included here for completeness.</u></p> <p>This is the UTRN Counter as carried in the locally entered 20 digit UTRN. It is the 10 least significant bits of the UTRN Counter, which is itself the 32 most significant bits of the Originator Counter for the Command.</p> <p>The PTUT truncated counter is not processed in WAN received top-up commands.</p>	<p>The PTUT Truncated Originator provides a means for a Device to derive the Originator Counter (and therefore the UTRN Counter) for the Prepayment Top Up Command when it is entered locally (as a numeric 20 digit code).</p> <p>In order to determine the UTRN Counter, the Device uses the algorithm defined in Section 14.6.</p>	<p>There is no additional impact to the Device as the same UTRN Counter cache is used as for the UTRN Counter.</p>
Remote Party Floor Sequence Numbers	<p>64-bit values carried in Update Security Credentials Command, in:</p> <ul style="list-style-type: none"> newRemotePartyFloorSeqNumber attribute; otherRemotePartyFloorSeqNumber sequence; newRemotePartySpecialistFloorSeqNumber attribute; and otherRemotePartySpecialistFloorSeqNumber sequence; <p>The values are used to set Counters associated with the credential being updated to new values. Processing is as detailed in</p>	<p>Remote Party Floor Sequence Numbers are of two types:</p> <ul style="list-style-type: none"> Remote Party Sequence Numbers. Values used to set Execution Counters on a change of a Remote Party's digital signing credential with which the counter is associated; and Remote Party Specialist Sequence Numbers. Value is used to populate UTRN Counter cache following its clearance on change of the Supplier Key Agreement Prepayment credential. <p>Both types have a 'new' and 'other' variant. 'new' is used when the authorising remote party is changing its own credentials (e.g.</p>	

	Section 13.3.5.10.	<p>supplier changing its own digital signing credential).</p> <p>'other' is used when the authorising remote party is changing the credentials of another remote party (e.g. TCoS changing supplier's credentials).</p> <p>Where – and only where - the Update Credentials Command changes the supplier entity ID (or indicates change of supplier), Counters are always reset – either to the Remote Party Sequence number indicated or to zero where the attribute is absent . Otherwise, Counters are only reset where the Remote Party Sequence Number is present.</p>	
Encryption Originator Counter	<u>The Counter value used for the purposes of encryption (see Section 8.3) for Responses and Alerts sent from the Device.</u>	<p>This is either the Supplementary Originator Counter in the case that this is to be included in the message (e.g. for an Unknown Remote Party) or the [Device] Originator Counter in all other instances.</p>	<p>The Device re-uses either the Supplementary Originator Counter or [Device] Originator Counter.</p>

6299

Table 23: Counters and their use in transaction identification and Protection Against Replay protection

6300 24 Annex 3 – ASN.1 modules - informative

6301 This Annex collates all ASN.1 used in this GBCS. Please note that this is a duplicate; the authoritative content remains as documented in the
6302 appropriate section.

```
6303 SetTime DEFINITIONS ::= BEGIN
6304
6305
6306 CommandPayload ::=          SEQUENCE
6307 {
6308     -- specify the period within which the Communications Hub's time must lie
6309     -- if this Command is successfully to set time
6310     validityIntervalStart           GeneralizedTime,
6311     validityIntervalEnd            GeneralizedTime
6312
6313 }
6314
6315 ResponsePayload ::=          SEQUENCE
6316
6317 {
6318     -- Specify the Device's now current time
6319     deviceTime                   GeneralizedTime,
6320
6321     -- Specify the Device's now current Time Status
6322     deviceTimeStatus              DeviceTimeStatus
6323 }
6324
6325 DeviceTimeStatus ::= INTEGER
6326 {
6327     reliable                    (0),
6328     invalid                     (1),
6329     unreliable                  (2)
6330 }
6331
6332 END
```

```
6333
6334 ActivateFirmware DEFINITIONS ::= BEGIN
6335
6336 CommandPayload ::=          SEQUENCE
6337 {
```

```
6338 -- specify the hash of the Manufacturer Image to be activated
6339 manufacturerImageHash          OCTET STRING,
6340
6341 -- the Originator Counter as in the Grouping Header of the Command
6342 originatorCounter           INTEGER (0..9223372036854775807),
6343
6344 -- the date-time at which the Command is to execute, if future dated
6345 executionDateTime           GeneralizedTime OPTIONAL
6346
6347 }
6348
6349 ResponsePayload ::= CHOICE
6350 {
6351   -- if the Command is future dated, the Response will not have any details of
6352   -- execution (those will be in the subsequent alert)
6353   commandAccepted           NULL,
6354
6355   -- if the Command is for immediate execution, the Response will detail the
6356   -- outcomes
6357   executionOutcome          ExecutionOutcome
6358
6359 }
6360
6361 AlertPayload ::= SEQUENCE
6362 {
6363   -- specify the Alert Code
6364   alertCode                 INTEGER(0..4294967295),
6365
6366   -- specify the date-time of execution
6367   executionDateTime          GeneralizedTime,
6368
6369   -- the Originator Counter as in the Grouping Header of the corresponding Command
6370   originatorCounter          INTEGER (0..9223372036854775807),
6371
6372   -- detail what happened when the future dated command was executed
6373   executionOutcome          ExecutionOutcome
6374
6375 }
6376
6377 ExecutionOutcome ::= SEQUENCE
6378 {
6379   -- Specify whether the activation was successful or not
6380   activateImageResponseCode  ActivateImageResponseCode,
```

```
6381
6382      -- Specify the Device's now current firmware version
6383      firmwareVersion          OCTET STRING
6384  }
6385
6386 ActivateImageResponseCode ::= INTEGER
6387 {
6388     success                  (0),
6389     noImageHeld              (1),
6390     hashMismatch             (2),
6391     activationFailure        (3)
6392 }
6393
6394 END
6395
6396 ProvideSecurityCredentialDetails DEFINITIONS ::= BEGIN
6397
6398 Command ::=           SEQUENCE
6399 {
6400   -- Identify which of the Public Keys on the Device is to be used in verifying the Signature or MAC
6401   -- (so defining the nature of the verification by way of the KeyUsage parameter held on the
6402   -- Device for the Public Key so identified).
6403
6404   authorisingRemotePartyTACellIdentifier      TrustAnchorCellIdentifier,
6405
6406   -- List the Remote Party Role(s) for which credential details are required
6407
6408   remotePartyRolesCredentialsRequired         SEQUENCE OF RemotePartyRole
6409 }
6410
6411 Response ::=           SEQUENCE OF RemotePartyDetails
6412
6413 RemotePartyDetails ::=      SEQUENCE
6414 {
6415
6416   -- Which Remote Party do these details relate to?
6417   remotePartyRole                   RemotePartyRole,
6418
6419   -- statusCode shall be success unless the role is not valid on this type of Device or there is a processing failure
6420   statusCode                         StatusCode,
6421
```

```
6422
6423 -- What is the current Update Security Credentials Protection Against Replay number on the Device for this role, where there is
6424 such a number for this role?
6425
6426 currentSeqNumber                               SeqNumber OPTIONAL,
6427
6428 -- What are the details held on the Device for each of the Cells related to this role? The list shall have between one and
6429 -- three entries (e.g. there will be one if role is transitional change of supplier; there may be three if role is supplier)
6430
6431 trustAnchorCellsDetails                      SEQUENCE OF TrustAnchorCellContents OPTIONAL
6432 }
6433
6434 SeqNumber ::=                                INTEGER (0..9223372036854775807)
6435
6436 TrustAnchorCellContents ::=                  SEQUENCE
6437 {
6438 -- To what cryptographic use can the Public Key in this Cell be put? Some Remote Party Roles
6439 -- (e.g. supplier) can have more than one Public Key on a Device and each one would only have
6440 -- a single cryptographic use.
6441
6442 trustAnchorCellKeyUsage                     KeyUsage,
6443
6444 -- trustAnchorCellUsage is to allow for multiple Public Keys of the same keyUsage for the same Remote
6445 -- Party Role. This will be absent except where used to refer to the Supplier Key Agreement Key.
6446 -- This Key is used solely in relation to validating Supplier generated MACs on Prepayment Top Up transactions.
6447
6448 trustAnchorCellUsage                         CellUsage DEFAULT management,
6449
6450 -- The subjectUniqueID which shall be the 64 bit Entity Identifier of the Security Credentials in this Trust Anchor Cell.
6451
6452 existingSubjectUniqueID                    OCTET STRING,
6453
6454 -- The APKI requirements mean that KeyIdentifier attributes will all be 8 byte SHA-1 Hashes.
6455 -- existingSubjectKeyIdentifier shall be set accordingly based on the contents of the Trust Anchor Cell
6456
6457 existingSubjectKeyIdentifier                OCTET STRING
6458 }
6459
6460 TrustAnchorCellIdentifier ::=             SEQUENCE
6461 {
6462 -- Which Remote Party Role does this Cell relate to?
6463
6464 trustAnchorCellRemotePartyRole            RemotePartyRole,
```

```
6465
6466 -- To what cryptographic use can the Public Key in this Cell be put? Some Remote Party Roles
6467 -- (e.g. supplier) can have more than one Public Key on a Device and each one would only have
6468 -- a single cryptographic use.
6469
6470 trustAnchorCellKeyUsage           KeyUsage,
6471
6472 -- trustAnchorCellUsage is to allow for multiple Public Keys of the same keyUsage for the same Remote
6473 -- Party Role. This may be absent except where use to refer to the Supplier Key
6474 -- Agreement Key used solely in relation to validating Supplier generated MACs on Prepayment Top Up transactions
6475
6476 trustAnchorCellUsage           CellUsage DEFAULT management
6477 }
6478
6479 CellUsage ::=                  INTEGER {management(0), prePaymentTopUp(1)}
6480
6481 RemotePartyRole ::=          INTEGER
6482 {
6483 -- Define the full set of Remote Party Roles in relation to which a Device may need to undertake
6484 -- processing. Note that most Devices will only support processing in relation to a subset of these.
6485
6486 root                      (0),
6487 recovery                   (1),
6488 supplier                   (2),
6489 networkOperator            (3),
6490 accessControlBroker        (4),
6491 transitionalCoS           (5),
6492 wanProvider                (6),
6493 issuingAuthority          (7),    -- Devices will receive such Certificates but they do not
6494 -- need to store them over an extended period
6495
6496
6497
6498 -- The 'other' RemotePartyRole is for a party whose role does not allow it to invoke any Device function apart from
6499 -- UpdateSecurityCredentials. This is to allow for Device functionality to be locked out of usage until a valid
6500 -- Remote Party can be identified e.g. where roles cannot be fixed until a Device is bought in to operation
6501 other                      (127)
6502
6503 }
6504
6505 -- KeyUsage is only repeated here for ease of reference. It is defined in RFC 5912
6506
6507 KeyUsage ::=                 BIT STRING
```

```
6508 {  
6509   -- Define valid uses of Public Keys.  
6510  
6511   digitalSignature          (0),  
6512   contentCommitment         (1),  -- not valid for GBCS compliant transactions  
6513   keyEncipherment          (2),  -- not valid for GBCS compliant transactions  
6514   dataEncipherment          (3),  
6515   keyAgreement              (4),  
6516   keyCertSign               (5),  
6517   cRLSign                   (6),  
6518   encipherOnly              (7),  
6519   decipherOnly              (8)   -- not valid for GBCS compliant transactions  
6520 }  
6521  
6522 -- The GBCS only allows for a constrained set of Trust Anchor Cell operations and so the list of possible outcomes  
6523 -- is more limited than in IETF RFC 5934. The list below is that more constrained subset  
6524  
6525 StatusCode ::= ENUMERATED {  
6526  
6527   success                  (0),  
6528  
6529   -- trustAnchorNotFound indicates that details of a trust anchor were requested, but the referenced trust anchor  
6530   -- is not represented on the Device  
6531  
6532   trustAnchorNotFound       (25),  
6533  
6534   other                     (127) }  
6535  
6536  
6537 END
```

```
6538  
6539 UpdateSecurityCredentials DEFINITIONS ::= BEGIN  
6540  
6541 CommandPayload ::= SEQUENCE  
6542 {  
6543   -- Provide details to allow the Device to identify the Remote Party Role authorising  
6544   -- this Command, check whether the rest of the payload is allowable, prevent replay attacks  
6545   -- and allow counters / counter caches on the Device to be reset, if the Command changes the Remote Party  
6546   -- in control.  
6547   -- The Remote Party authorising the Command is that party which generated the KRP Signature (or the Access Control Broker  
6548   -- if there is no KRP Signature)
```

```
6549
6550 authorisingRemotePartyControl          AuthorisingRemotePartyControl,
6551
6552 -- One TrustAnchorReplacement structure is required for each Trust Anchor Cell that is to be updated
6553
6554 replacements                      SEQUENCE OF TrustAnchorReplacement,
6555
6556 -- Provide the certificates needed to undertake Certification Path Validation of the new
6557 -- end entity certificate against the root public key held on the Device. The number of these may be less
6558 -- than the number of replacement certificates (e.g. a supplier may replace all of its certificates but
6559 -- may only need to supply one Certification Authority Certificate to link them all back to the root public
6560 -- key as currently stored on the Device.
6561
6562 certificationPathCertificates      SEQUENCE OF Certificate,
6563
6564 -- If the Command is to be future dated, specify the date-time at which the certificate replacement is to happen
6565
6566 executionDateTime                  GeneralizedTime OPTIONAL
6567
6568 }
6569
6570 ResponsePayload ::=           SEQUENCE
6571 {
6572   -- if the Command is future dated, the Response will not have any details of execution (those will be in the subsequent alert)
6573
6574   commandAccepted                 NULL,
6575
6576   -- if the Command is for immediate execution, the Response will detail the outcomes
6577
6578   executionOutcome                ExecutionOutcome OPTIONAL
6579
6580 }
6581
6582 AlertPayload ::=             SEQUENCE
6583 {
6584   -- specify the Alert Code
6585   alertCode                      INTEGER(0..4294967295),
6586
6587   -- specify the date-time of execution
6588   executionDateTime               GeneralizedTime,
6589
6590
6591   -- detail what happened when the future dated Command was executed
```

```
6592     executionOutcome           ExecutionOutcome
6593
6594 }
6595
6596 ExecutionOutcome ::=          SEQUENCE
6597 {
6598   -- Provide details of the corresponding Command that may not be in the standard GBCS message header. Specifically the
6599   -- mode in which the Command was invoked, the Originator Counter in the original Command and the resulting changes to any
6600   -- replay counters held on the Device
6601
6602   authorisingRemotePartySeqNumber      SeqNumber,
6603   credentialsReplacementMode          CredentialsReplacementMode,
6604   remotePartySeqNumberChanges        SEQUENCE OF RemotePartySeqNumberChange,
6605
6606   -- For each replacement in the Command, detail the outcome and impacted parties
6607
6608   replacementOutcomes               SEQUENCE OF ReplacementOutcome
6609
6610 }
6611
6612 AuthorisingRemotePartyControl ::=    SEQUENCE
6613 {
6614   -- Specify the replacement mode so that the Device can check that the Remote Party Role is allowed to
6615   -- authorise this type of replacement and that all replacements in the payload are allowed within this
6616   -- replacement mode
6617
6618   credentialsReplacementMode          CredentialsReplacementMode,
6619
6620   -- Only if credentialsReplacementMode = anyByContingency, provide the symmetric key to decrypt
6621   -- the Contingency Public Key in the (root, digitalSignature, management) Trust Anchor Cell
6622
6623   plaintextSymmetricKey              [0] IMPLICIT OCTET STRING OPTIONAL,
6624
6625   -- Specify whether the time based checks as part of any Certificate Path Validation should be applied
6626
6627   applyTimeBasedCPVChecks           [1] IMPLICIT INTEGER {apply(0), disapply(1)} DEFAULT apply,
6628
6629   -- Identify which of the Public Keys on the Device is to be used in checking KRP Signature
6630   -- 'authorisingRemotePartyTACellIdentifier' can only be omitted when
6631   -- the access control broker is updating its own credentials. In all other cases it is mandatory.
6632
6633   authorisingRemotePartyTACellIdentifier [2] IMPLICIT TrustAnchorCellIdentifier OPTIONAL,
```

```
6635
6636 -- Specify the Originator Counter for the Remote Party Applying KRP Signature, or (for the
6637 -- Access Control Broker changing its credentials) the Access Control Broker's Originator Counter.
6638
6639 authorisingRemotePartySeqNumber           [3] IMPLICIT SeqNumber,
6640
6641 -- If the Command is to effect a change of control, then newTrustAnchorFloorSeqNumber must be included
6642 -- and will be the value used to prevent replay of Update Security Credentials Commands for the
6643 -- new controlling Remote Party.
6644
6645 newRemotePartyFloorSeqNumber           [4] IMPLICIT SeqNumber OPTIONAL,
6646
6647 -- Some Commands on the Device may use a different Originator Counter sequence for Protection Against Replay. At this
6648 -- version of the GBCS, the only example is the Prepayment Top Up Command on ESME and GSME. The
6649 -- SpecialistSeqNumber structure allows such Counters to also be reset on change of control.
6650
6651 newRemotePartySpecialistFloorSeqNumber      [5] IMPLICIT SEQUENCE OF SpecialistSeqNumber OPTIONAL,
6652
6653 -- In some cases, one party acting in one Remote Party Role may be replacing certificates for a different Remote Party Role.
6654 -- In some cases, sequence counters need also to be reset for those other Remote Party Role(s)
6655
6656 otherRemotePartySeqNumberChanges          [6] IMPLICIT SEQUENCE OF RemotePartySeqNumberChange OPTIONAL
6657 }
6658
6659 RemotePartySeqNumberChange ::=           SEQUENCE
6660 {
6661   otherRemotePartyRole                  RemotePartyRole,
6662   otherRemotePartyFloorSeqNumber        SeqNumber,
6663   newRemotePartySpecialistFloorSeqNumber SEQUENCE OF SpecialistSeqNumber OPTIONAL
6664 }
6665
6666 SpecialistSeqNumber ::=                SEQUENCE
6667 {
6668   -- Specify the usage of the SeqNumber
6669   seqNumberUsage                     SeqNumberUsage,
6670
6671   -- Specify the associated SeqNumber
6672   seqNumber                         SeqNumber
6673 }
6674
6675 SeqNumberUsage ::=                      INTEGER
6676 {
6677   -- Define the full set of discrete usages on a Device. The only specialist
```

```
6678 -- counter is for Prepayment Top Up (which is set independently of other counters). This may only be
6679 -- included when changing Supplier Security Credentials on an ESME or GSME.
6680
6681 prepaymentTopUp          (0)
6682 }
6683
6684 SeqNumber ::=           INTEGER (0..9223372036854775807)
6685
6686
6687 TrustAnchorReplacement ::=      SEQUENCE
6688 {
6689   -- Provide the new end entity certificate
6690
6691   replacementCertificate      Certificate,
6692
6693   -- Specify where it is to go (specifically which Trust Anchor Cell is to have its details replaced using
6694   -- the new end entity certificate)
6695
6696   targetTrustAnchorCell       TrustAnchorCellIdentifier
6697 }
6698
6699
6700 ReplacementOutcome ::=      SEQUENCE
6701 {
6702   affectedTrustAnchorCell    TrustAnchorCellIdentifier,
6703   statusCode                StatusCode,
6704
6705   -- The GBCS Certificate requirements mean that the subjectUniqueID attribute in the subject field of a certificate will always
6706   -- contain the 64 bit unique number that equates to Entity Identifier. existingSubjectUniqueID should be set
6707   -- accordingly based on the contents of the Trust Anchor Cell prior to Command processing.
6708
6709   existingSubjectUniqueID    OCTET STRING,
6710
6711   -- The GBCS Certificate requirements mean that subjectKeyIdentifier attributes will all be 8 byte SHA-1 Hashes.
6712   -- existingSubjectKeyIdentifier should be set accordingly based on the contents of the Trust Anchor Cell prior to
6713   -- Command processing.
6714
6715   existingSubjectKeyIdentifier OCTET STRING,
6716
6717   -- The subjectUniqueID in the subject field of the certificate in this TrustAnchorReplacement
6718
6719   replacingSubjectUniqueID   OCTET STRING,
6720
```

```
6721 -- The subjectKeyIdentifier in the certificate in this TrustAnchorReplacement
6722
6723 replacingSubjectKeyIdentifier          OCTET STRING
6724 }
6725
6726 TrustAnchorCellIdentifier ::=           SEQUENCE
6727 {
6728   -- Which Remote Party Role does this Cell relate to?
6729
6730   trustAnchorCellRemotePartyRole        RemotePartyRole,
6731
6732   -- To what cryptographic use can the Public Key in this Cell be put? Some Remote Party Roles
6733   -- (e.g. supplier) can have more than one Public Key on a Device and each one would only have
6734   -- a single cryptographic use.
6735
6736   trustAnchorCellKeyUsage              KeyUsage,
6737
6738   -- trustAnchorCellUsage is to allow for multiple Public Keys of the same keyUsage for the same Remote
6739   -- Party Role. It will be absent except where used to refer to the Supplier Key
6740   -- Agreement Key used solely in relation to validating Supplier generated MACs on Prepayment Top Up
6741   -- transactions
6742
6743   trustAnchorCellUsage                CellUsage DEFAULT management
6744 }
6745
6746 CellUsage ::=                         INTEGER {management(0), prePaymentTopUp(1)}
6747
6748 RemotePartyRole ::=                  INTEGER
6749 {
6750   -- Define the full set of Remote Party Roles in relation to which a Device may need to undertake
6751   -- processing. Note that most Devices will only support a subset of these.
6752
6753   root                      (0),
6754   recovery                   (1),
6755   supplier                  (2),
6756   networkOperator            (3),
6757   accessControlBroker        (4),
6758   transitionalCoS            (5),
6759   wanProvider                (6),
6760   issuingAuthority           (7),    -- Devices will receive such Certificates but they do not need to store
6761                                -- them over an extended period
6762
6763   -- The 'other' RemotePartyRole is for a party whose role does not allow it to invoke any Device function apart from
```

```
6764 -- UpdateSecurityCredentials. This is to allow for Device functionality to be locked out of usage until a valid
6765 -- Remote Party can be identified e.g. where roles cannot be fixed until a Device is brought in to operation
6766
6767 other                                (127)
6768 }
6769
6770
6771 -- KeyUsage is only repeated here for clarity. It is defined in RFC 5912
6772
6773 KeyUsage ::=                      BIT STRING
6774 {
6775   -- Define valid uses of Public Keys held by Devices in their Trust Anchor Cells.
6776
6777   digitalSignature                  (0),
6778   contentCommitment                (1),    -- not valid for GBCS compliant transactions
6779   keyEncipherment                 (2),    -- not valid for GBCS compliant transactions
6780   dataEncipherment                (3),    -- not valid for GBCS compliant transactions
6781   keyAgreement                   (4),
6782   keyCertSign                     (5),
6783   cRLSign                         (6),
6784   encipherOnly                    (7),    -- not valid for GBCS compliant transactions
6785   decipherOnly                   (8)     -- not valid for GBCS compliant transactions
6786 }
6787
6788 CredentialsReplacementMode ::=      INTEGER
6789 {
6790   -- Define the valid combinations as to which Remote Party Roles can replace which kinds of Trust Anchors.
6791
6792   -- Normal operational replacement modes
6793   rootBySupplier                  (0),
6794   rootByWanProvider               (1),
6795   supplierBySupplier              (2),
6796   networkOperatorByNetworkOperator (3),
6797   accessControlBrokerByACB        (4),
6798   wanProviderByWanProvider       (5),
6799   transCoSByTransCoS             (6),
6800   supplierByTransCoS            (7),
6801
6802   -- Recovery modes
6803   anyExceptAbnormalRootByRecovery (8),
6804   anyByContingency                (9)
6805 }
6806
```

```
6807 -- The GBCS only allows for a constrained set of Trust Anchor Cell operations and so the list of possible outcomes
6808 -- is more limited than in RFC 5934. The list below is that more constrained subset
6809
6810 StatusCode ::= ENUMERATED {
6811   success                      (0),
6812
6813   -- badCertificate is used to indicate that the syntax for one or more certificates is invalid.
6814   badCertificate                 (5),
6815
6816   -- noTrustAnchor is used to indicate that the authorityKeyIdentifier does not identify the public key of a
6817   -- trust anchor or a certification path that terminates with an installed trust anchor
6818
6819   noTrustAnchor                  (10),
6820
6821   -- insufficientMemory indicates that the update could not be processed because the Device did not
6822   -- have sufficient memory
6823
6824   insufficientMemory            (17),
6825
6826   -- contingencyPublicKeyDecrypt indicates that the update could not be processed because an error occurred while
6827   -- decrypting the contingency public key.
6828
6829   contingencyPublicKeyDecrypt    (22),
6830
6831   -- trustAnchorNotFound indicates that a change to a trust anchor was requested, but the referenced trust anchor
6832   -- is not represented in the Trust Anchor Cell.
6833
6834   trustAnchorNotFound           (25),
6835
6836   -- resourcesBusy indicates that the resources necessary to process the replacement are not available at the
6837   -- present time, but the resources might be available at some point in the future.
6838
6839   resourcesBusy                 (30),
6840
6841   -- other indicates that the update could not be processed, but the reason is not covered by any of the assigned
6842   -- status codes. Use of this status code SHOULD be avoided.
6843
6844   other                         (127) }
```

6849
6850 IssueSecurityCredentials DEFINITIONS ::= BEGIN
6851
6852 CommandPayload ::= SEQUENCE
6853 {
6854 -- specify the keyUsage to which the generated key-pair will be put, if subsequently authorised
6855 keyUsage
6856
6857 }
6858
6859 ResponsePayload ::= CHOICE
6860
6861 {
6862 -- if the Command was successful, provide the generated Certification Request. CertificationRequest shall
6863 -- be as defined in ASN.1 by IETF RFC 5912. For reference, it is in the section headed 'ASN.1 Module for RFC 2986'
6864 certificationRequest
6865
6866 -- if the Command was unsuccessful, detail the failure
6867
6868 issueCredentialsResponseCode
6869 }
6870
6871 -- KeyUsage is only repeated here for ease of reference. It is defined in IETF RFC 5912
6872
6873 KeyUsage ::= BIT STRING
6874 {
6875 -- Define valid uses of Public Keys held by Devices in their Trust Anchor Cells.
6876
6877 digitalSignature
6878 contentCommitment
6879 keyEncipherment
6880 dataEncipherment
6881 keyAgreement
6882 keyCertSign
6883 cRLSign
6884 encipherOnly
6885 decipherOnly
6886 }
6887
6888 IssueCredentialsResponseCode ::= INTEGER
6889 {
6890 invalidKeyUsage
6891 (1),

```
6891     keyPairGenerationFailed          (2),
6892     cRProductionFailed              (3)
6893 }
6894
6895
6896 END
6897
6898 UpdateDeviceCertificateonDevice DEFINITIONS ::= BEGIN
6899
6900 CommandPayload ::=                  Certificate
6901   -- provide the certificate which the Device is to store
6902   -- the ASN.1 specification of certificate shall be as defined in IETF RFC 5912 for IETF RFC 5280
6903
6904 ResponsePayload ::=                 UpdateDeviceCertResponseCode
6905
6906   -- if the Command was unsuccessful, detail the failure; otherwise respond with success
6907
6908 UpdateDeviceCertResponseCode ::=      INTEGER
6909 {
6910   success                      (0),
6911   invalidCertificate           (1),
6912   wrongDeviceIdentity          (2),
6913   invalidKeyUsage              (3),
6914   noCorrespondingKeyPairGenerated (4),
6915   wrongPublicKey                (5),
6916   certificateStorageFailed     (6),
6917   privateKeyChangeFailed       (7)
6918 }
6919
6920 END
6921
6922 ProvideDeviceCertificateFromDevice DEFINITIONS ::= BEGIN
6923
6924 CommandPayload ::=                  SEQUENCE
6925 {
6926   -- specify the KeyUsage of the Certificate to be returned
6927   keyUsage                     KeyUsage
6928 }
6929
6930
```

```
6931 ResponsePayload ::= CHOICE
6932 {
6933     -- if the Command was successful, provide the certificate
6934     certificate
6935             Certificate,
6936
6937     -- if the Command was unsuccessful, detail the failure
6938
6939     provideDeviceCertResponseCode      ProvideDeviceCertResponseCode
6940 }
6941
6942
6943 -- KeyUsage is only repeated here for ease of reference. It is defined in RFC 5912
6944
6945 KeyUsage ::= BIT STRING
6946 {
6947     -- Define valid uses of Public Keys held by Devices in their Trust Anchor Cells.
6948
6949     digitalSignature          (0),
6950     contentCommitment        (1),    -- not valid for GBCS compliant transactions
6951     keyEncipherment          (2),    -- not valid for GBCS compliant transactions
6952     dataEncipherment         (3),    -- not valid for GBCS compliant transactions
6953     keyAgreement            (4),
6954     keyCertSign              (5),    -- not valid for this Use Case
6955     cRLSign                 (6),    -- not valid for this Use Case
6956     encipherOnly             (7),    -- not valid for GBCS compliant transactions
6957     decipherOnly             (8)     -- not valid for GBCS compliant transactions
6958 }
6959
6960 ProvideDeviceCertResponseCode ::= INTEGER
6961 {
6962     invalidKeyUsage          (1),
6963     noCertificateHeld        (2),
6964     certificateRetrievalFailure (3)
6965 }
6966
6967
6968 END
6969
6970 JoinDevice DEFINITIONS ::= BEGIN
6971
```

```
6972 CommandPayload ::= SEQUENCE
6973 {
6974     -- specify which type of joining is being authorised and,
6975     -- for Method A Joins, the role the Device is to play
6976
6977     joinMethodAndRole           JoinMethodAndRole,
6978
6979     -- specify the Entity Identifier of the Device which is to be Joined with
6980
6981     otherDeviceEntityIdentifier OCTET STRING,
6982
6983     -- specify the DeviceType of that other Device
6984
6985     otherDeviceType             DeviceType,
6986
6987     -- provide the other Device's Key Agreement certificate, if and only if this
6988     -- is a join between a gSME and a type1PrepaymentInterfaceDevice.
6989     -- Certificate shall be as defined in IETF RFC 5912
6990
6991     otherDeviceCertificate      Certificate OPTIONAL
6992
6993 }
6994
6995 -- detail whether the Command successful executed or, if it didn't,
6996 -- what the failure reason was
6997
6998 ResponsePayload ::= JoinResponseCode
6999
7000 JoinMethodAndRole ::= INTEGER
7001 {
7002     -- methodB is to be used where the other Device is a Type 2 Device or GPF.
7003     -- methodC is used where the Devices involved are a GSME and a PPMID.
7004     -- methodA is used otherwise.
7005     -- methodAInitiator is used where the Device this Command is targeted at
7006     -- should initiate the Key Agreement process
7007     -- methodAResponder is used where the Device this Command is targeted at
7008     -- should respond in the Key Agreement process, but shall not initiate it
7009
7010     methodAInitiator          (0),
7011     methodAResponder          (1),
7012     methodB                  (2),
7013     methodC                  (3)
7014 }
```

```
7015
7016 DeviceType ::= INTEGER
7017 {
7018     gSME          (0),
7019     eSME          (1),
7020     communicationsHubCommunicationsHubFunction (2),
7021     communicationsHubGasProxyFunction      (3),
7022     type1HANConnectedAuxiliaryLoadControlSwitch (4),
7023     type1PrepaymentInterfaceDevice        (5),
7024     type2           (6)
7025 }
7026
7027 JoinResponseCode ::= INTEGER
7028 {
7029     success          (0),
7030     invalidMessageCodeForJoinMethodAndRole (1),
7031     invalidJoinMethodAndRole      (2),
7032     incompatibleWithExistingEntry   (3),
7033     deviceLogFull        (4),
7034     writeFailure         (5),
7035     keyAgreementNoResources    (6),
7036     keyAgreementUnknownIssuer   (7),
7037     keyAgreementUnsupportedSuite (8),
7038     keyAgreementBadMessage     (9),
7039     keyAgreementBadKeyConfirm  (10),
7040     invalidOrMissingCertificate (11)
7041 }
7042
7043 END
7044
7045 UnjoinDevice DEFINITIONS ::= BEGIN
7046
7047 CommandPayload ::= OtherDeviceEntityIdentifier
7048     -- specify the Entity Identifier of the Device for which authorisation
7049     -- is to be removed
7050
7051     OtherDeviceEntityIdentifier ::= OCTET STRING
7052
7053 ResponsePayload ::= UnjoinResponseCode
7054
7055     -- detail whether the Command successful executed or, if it didn't,
```

```
7056      -- what the failure reason was
7057
7058 UnjoinResponseCode ::= INTEGER
7059 {
7060     success                      (0),
7061     otherDeviceNotInDeviceLog    (1),
7062     otherFailure                 (2)
7063 }
7064
7065 END
7066
7067 ReadDeviceLog DEFINITIONS ::= BEGIN
7068
7069 CommandPayload ::= NULL
7070
7071 ResponsePayload ::= SEQUENCE
7072 {
7073     -- detail whether the Command successful
7074
7075     readLogResponseCode          ReadLogResponseCode,
7076
7077     -- if it was, return the Log Entries
7078     deviceLogEntries             SEQUENCE OF DeviceLogEntry OPTIONAL
7079 }
7080
7081 DeviceLogEntry ::= SEQUENCE
7082 {
7083     deviceIdentifier            OCTET STRING,
7084     deviceType                  DeviceType
7085 }
7086
7087 DeviceType ::= INTEGER
7088 {
7089     gSME                        (0),
7090     eSME                        (1),
7091     communicationsHubCommunicationsHubFunction (2),
7092     communicationsHubGasProxyFunction   (3),
7093     type1HANConnectedAuxiliaryLoadControlSwitch (4),
7094     type1PrepaymentInterfaceDevice     (5),
7095     type2                        (6)
7096 }
```

```
7097  
7098  
7099 ReadLogResponseCode ::= INTEGER  
7100 {  
7101   success          (0),  
7102   readFailure      (1)  
7103 }  
7104  
7105 END  


---

  
7106  
7107 GPFDeviceLog DEFINITIONS ::= BEGIN  
7108  
7109 BackupAlertPayload ::= SEQUENCE  
7110 {  
7111   -- specify the Alert Code  
7112   alertCode          INTEGER(0..4294967295),  
7113  
7114   -- specify the date-time of the backup  
7115   backupDateTime     GeneralizedTime,  
7116  
7117   -- detail the entries in the Device Log now that the change has been made  
7118   deviceLogEntries    SEQUENCE OF DeviceLogEntry  
7119  
7120 }  
7121  
7122 RestoreCommandPayload ::= SEQUENCE  
7123 {  
7124   -- list the Device Log entries that are to be added  
7125   deviceLogEntries    SEQUENCE OF DeviceLogEntry  
7126  
7127 }  
7128  
7129 DeviceLogEntry ::= SEQUENCE  
7130 {  
7131  
7132   -- specify the Entity Identifier of the Device  
7133   deviceEntityIdentifier OCTET STRING,  
7134  
7135   -- specify the DeviceType of that Device  
7136  
7137   deviceType          DeviceType
```

```
7138 }
7139
7140 RestoreResponsePayload ::= SEQUENCE
7141
7142 {
7143     -- for each DeviceLog Entry, detail whether the Command successfully executed or, if it didn't, what the failure reason was
7144
7145     restoreOutcomes           SEQUENCE OF RestoreOutcome
7146 }
7147
7148 RestoreOutcome ::= SEQUENCE
7149 {
7150     deviceLogEntry            DeviceLogEntry,
7151     joinResponseCode          JoinResponseCode
7152 }
7153
7154 DeviceType ::= INTEGER
7155 {
7156     gSME                      (0),
7157     eSME                      (1),
7158     communicationsHubFunction (2),
7159     communicationsHubGasProxyFunction (3),
7160     type1HANConnectedAuxiliaryLoadControlSwitch (4),
7161     type1PrepaymentInterfaceDevice (5),
7162     type2                      (6)
7163 }
7164
7165 JoinResponseCode ::= INTEGER
7166 {
7167     success                  (0),
7168     invalidMessageCodeForJoinMethodAndRole (1),
7169     invalidJoinMethodAndRole (2),
7170     incompatibleWithExistingEntry (3),
7171     deviceLogFull             (4),
7172     writeFailure              (5),
7173     keyAgreementNoResources   (6),
7174     keyAgreementUnknownIssuer (7),
7175     keyAgreementUnsupportedSuite (8),
7176     keyAgreementBadMessage    (9),
7177     keyAgreementBadKeyConfirm (10),
7178     invalidOrMissingCertificate (11)
7179 }
7180 }
```

7181

7182 END

7183

7184 **25 Annex 4 - Use of ZigBee in GBCS -** 7185 **informative**

7186 **25.1 Purpose**

7187 This annex briefly summarises where the GBCS:

- 7188 • requires the use of ZigBee, specifically where it uses parts of the ZigBee specification,
7189 or takes an approach which aligns to the ZigBee specification; and
- 7190 • does not allow the use of ZigBee / requires its use to be modified, specifically where it:
 - 7191 ○ mandates a solution that is not ZigBee derived but where there is ZigBee equivalent
7192 in the specification;
 - 7193 ○ specifies an approach that is derived from ZigBee but the approach is not part of the
7194 ZigBee specification; and
 - 7195 ○ specifies an approach that uses parts of the ZigBee but varies from it on specific
7196 points.

7197 The document is based on the content of ZigBee referenced in the GBCS.

7198 **25.2 GBCS requirements to use ZigBee**

7199 For all Smart Metering Equipment, the GBCS requires the implementation of functionality
7200 equivalent to a subset of the ZigBee standard, including all mandatory components required
7201 to achieve ZigBee certification.

7202 GBCS and the ZigBee standard specify all items that need to be certified. GBCS does not
7203 require ZSE certification for non-standard ZSE features in Smart Metering Equipment.

7204 **25.3 GBCS requirements not to use ZigBee / vary from it**

7205 For GSME and PPMID, the GBCS requires functionality equivalent to ZigBee clusters, but
7206 transports GBZ payloads using ZigBee tunneling.

7207 GBCS does not require ZSE certification for non-ZSE features in GBCS. End-to-end
7208 Messages sent with ZigBee (ZSE / ZCL) commands are referred to as GBZ.

26 Annex 5 - Use of DLMS COSEM in GBCS - informative

26.1 Purpose

7212 This document briefly summarises where the GBCS:

- 7213 • requires the use of DLMS COSEM: specifically where it uses parts of the DLMS
7214 COSEM specification, or takes an approach which aligns to the DLMS COSEM
7215 specification; and
- 7216 • does not allow the use of DLMS COSEM / requires its use to be modified: specifically
7217 where it:
 - 7218 ○ Mandates a solution that is not DLMS COSEM derived but where there are DLMS
7219 COSEM equivalent in the specification;
 - 7220 ○ Specifies an approach that is derived from DLMS COSEM but the approach is not
7221 part of the DLMS COSEM specification; or
 - 7222 ○ Specifies an approach that uses parts of the DLMS COSEM but varies from it on
7223 specific points.

7224 The document is based on the expected contents of the Blue and Green Book versions
7225 scheduled to be published in June / July 2014.

26.2 GBCS requirements to use DLMS COSEM

7227 For ESME and CHF, the GBCS requires the implementation of functionality equivalent to a
7228 subset of the Blue Book Classes. It does not require functionality equivalent to other Blue
7229 Book classes.

7230 For all Devices, GBCS requires a set of cryptographic primitives that align to DLMS Security
7231 Suite 1, and so all Devices will need functionality which is in line with the cryptography
7232 related parts of the Green Book (for both GBCS and DLMS COSEM, these requirements are
7233 NSA Suite B derived).

7234 GBCS requires that all Devices use X.509 Certificates and Certification Requests with a
7235 number of optional elements being used / barred. These requirements align with the Green
7236 Book requirements (which are X.509 derived).

7237 For ESME and CHF, the GBCS requires functionality equivalent to Green Book access and
7238 data notification services.

7239 For all Devices, the GBCS requires functionality equivalent to the Green Book's general
7240 ciphering and general signing services.

7241 For all Devices, the GBCS requires functionality equivalent to the Green Book's
7242 authenticated encryption and decryption.

7243 For all Devices, the GBCS requires corresponding alignment with DLMS COSEM's ASN.1
7244 schema and its A-XDR encoding.

26.3 GBCS requirements not to use DLMS COSEM / vary from it

26.3.1 Mandates a solution that is not DLMS COSEM derived but where there are DLMS COSEM equivalent in the specification

For Devices other than ESME and CHF, the GBCS requires functionality equivalent to DLMS COSEM classes but does not use DLMS COSEM classes (rather ZSE / ASN.1 is used).

For Devices other than ESME and CHF, the GBCS requires support for equivalents of the Green Book's access and data notification services, but uses ZSE or ASN.1 specific structures.

For all Devices, the GBCS requires that the management of X.509 certificates and Device's key pairs is undertaken using ASN.1 messages derived from the IETF's TAMP RFCs.

Over the HAN, the GBCS mandates, for all Devices, the use of ZSE for the communication layers below the DLMS/COSEM Application Layer and so does not allow the use of the equivalent Green Book communication profiles. (WAN transport is outside GBCS scope).

For ESME and GSME, distribution of firmware is through the ZSE OTA mechanism.

26.3.2 Specifies an approach that is derived from DLMS COSEM but the approach is not part of the specification

GBCS specifies the use of a Class 9000 object. This is not in the Blue Book

Although not yet incorporated, the proposals to use the DLMS Blocking Service would see DLMS type structures being used in a way not specified in the DLMS COSEM specification.

Pairwise key agreement between GSME and PPMID uses a structure similar to DLMS COSEM's message structure, but that is not part of the DLMS COSEM specification.

26.3.3 Specifies an approach that uses parts of the DLMS COSEM but varies from it on specific points

For all bar Type 2 Devices, the DLMS general-signing structure is used in all remote party messages but the signature field is not populated in messages that do not require a signature (i.e. those that are not critical).

For all bar Type 2 Devices, the GBCS uses the general-ciphering structure for remote party Messages that require a MAC. The GBCS leaves most values empty in the header part of the structure (these value are either in the general-signing structure or are already known to the meter). Correspondingly, the values used in the OtherInput field of the KDF at Section 9.2.3.4.6.5 of the Green Book are those taken from the general-signing structure, rather than the corresponding fields in the general-ciphering structure.

For ESME and GSME, the GBCS specifies particular, additional interpretation of parameters within the DLMS COSEM Class 8 object (Clock).

7280 27 Annex 6 - Deducing the UTRN Counter from the 7281 Truncated UTRN Counter – informative

This annex provides a worked example of the calculation described in Section 14.6.4.1.5. The calculation uses the 10-bit Truncated UTRN Counter received with the prepay top-up command is received via Consumer Entry to the Device, either directly or via a PPMID. The calculation uses the highest UTRN Counter value held in the Device's UTRN Counter cache, and a window of 512 either side of this value in making the deduction.

7286 In this case, the UTRN Counter being entered into the Device is 5 greater than the highest thus far received by the Device.

Step	Description	Example	
		Binary Representation	Decimal Representation
1	The method requires 4 signed 32 bit integers, p , q , r and s		
2	Set $p =$ the numeric value of the least significant 10 bits of the highest UTRN Counter value in the UTRN Counter cache (V)	1100100111	807
3	Set $q = V - p$ $q = 2,458,896,167 - 807$	10010010100011111000100000 00000	2,458,895,360

4	Set $r = \text{PTUT Truncated Originator Counter}$	1100101100	812
5	Calculate $p - 2^9$ (Call this variable, x) <small>(See footnote 39)</small> $x = 812 - 512$	100101100	300
6	Calculate $p + 2^9$ (Call this variable, y) $y = 812 + 512$	10100101100	1324
7	Test r against x and y and set s accordingly <ul style="list-style-type: none"> • If $r < x$ then $s = r + 2^{10}$ • If $r > y$ then $s = r - 2^{10}$ • Else $s = r$ 300 < 812 < 1324, therefore $s = r$	1100101100	812
8	Set deduced Originator Counter = $(q + s) * 2^{32}$	10010010100011111000111001 01100000000000000000000000000000 0000000000	10,560,878,642,999,590,912
9	Set deduced UTRN Counter as most significant 32 bits of Deduced Originator Counter	10010010100011111000111001 01100	2,458,896,172

Table 27: Derivation of the UTRN Counter from the PTUT Truncated UTRN Counter – a worked example

³⁹ In some cases where $p < 512$, this result may be negative. How negative binary numbers are represented in the calculation is an implementation decision, and not a matter for the GBCS since there is no impact on interoperability.

7288

Crown copyright 2014

Department of Energy & Climate Change
3 Whitehall Place
London SW1A 2AW

www.gov.uk/decc

URN 14D/439