Smart Metering Implementation Programme

Great Britain Companion Specification (GBCS)

Version 4.~~0~~1 (MP164 changes only, drafted against the approved SECMP0007 GBCS legal text)

~~29 November 2020~~xx November 2021

This page is left blank.

Documentation Alignment

DLMS Green and Blue Books

This document aligns with the published versions of the Green Book (DLMS UA 1000-2 Ed. 8) 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

This document aligns with:

- ZigBee Smart Energy (ZSE) Profile Specification 1.4;
- ZigBee Cluster Library Specification, reference 07-5123-04;
- ZigBee OTA Upgrade Cluster Specification, reference 09-5264-23;
- ZigBee Specification Revision 22, reference 05-3474-22; and
- ZigBee PRO/2007 Layer PICS and Stack Profiles – 08-0006-05.

These documents are available from http://zigbee.org/About/GBCSPartner.aspx or by contacting the ZigBee Alliance.

Table of Contents

1 Introduction – normative 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 14

4.1 Introduction – informative 14
4.2 Cryptographic Protections applying to all Messages 15
4.3 Security for Remote Party Messages 15

5 Remote Party Message construction, protection and verification – informative 29

5.1 Common Message Structures – informative 29
5.2 Common Encryption and Decryption approach – informative 29
5.3 Message Categories – informative 29
5.4 Common Message Processing steps – informative 30

6 Message Categories 33

6.1 Introduction – informative 33
6.2 Message Category SME.C 33
6.3 Message Category SME.C.C 38
6.4 Message Category SME.C.NC 39
6.5 Message Category SME.C.PPMID-GSME 41
6.6 Message Category SME.A 43
6.7 Message Category SME.A.C 44
6.8 Message Category SME.A.NC 44

7 Message structure and DLMS COSEM / ZSE / ASN.1 requirements 47

7.1 Introduction – informative 47
7.2 Remote Party Message construction – general 47
7.3 Device Requirements – DLMS COSEM 69
7.4 Device requirements – ZSE 80

8 Encryption of Attributes in Remote Party Messages 83

8.1 Approach – informative 83
8.2 Common requirements 83
8.3 Key Derivation Inputs 84
8.4 AAD, Plaintext and Ciphertext 84
8.5 Access to sensitive data – COSEM attribute access 85
8.6 Access to sensitive data – ZSE attribute access 92

9 Time Synchronisation and Future Dated Remote Party Messages 93

9.1 Time synchronisation 93
9.2 Future Dated Remote Party Messages 100

10 ZSE Implementation 106

10.1 Introduction – informative ... 106
10.2 Tunnels ... 106
10.3 GSME and GPF interactions ... 114
10.4 GPF Structured Data Items ... 117
10.5 Hand Held Terminal (HHT) interactions ... 123
10.6 Sub GHz Requirements ... 129
10.7 SMHAN management ... 144
10.8 CH routing of Remote Party and SME.C.PPMID-GSME Messages ... 147

11 Downloading firmware images to Devices ... 149

11.1 Introduction – informative ... 149
11.2 Common Requirements ... 150
11.3 CS05a Distribute Firmware to Communications Hub ... 154
11.4 CS05b Distribute Firmware to ESME / GSME / HCALCS ... 154
11.5 CS06 Activate Firmware ... 157
11.6 CS05c Distribute Firmware to PPMID ... 160
11.7 CCS08 Firmware Transfer Alert ... 162
11.8 CS08 Read PPMID/HCALCS Firmware Version ... 164

12 Requirements for Certificates ... 170

12.1 Requirements applicable to all Certificates ... 170
12.2 Requirements applicable to Organisation Certificates only ... 171
12.3 Requirements applicable to Certificates where `RemotePartyRole = root` or `issuingAuthority` ... 171
12.4 Requirements applicable to Certificates where `RemotePartyRole` is neither `root` nor `issuingAuthority` ... 172
12.5 Requirements applicable to Device Certificates ... 172
12.6 Device processing of Certificates ... 173

13 Managing Security Credentials on Devices ... 174

13.1 Introduction – informative ... 174
13.2 CS02a Provide Security Credential Details Command and Response ... 176
13.3 CS02b Update Security Credentials Command, Response and Alert ... 190
13.4 CS02c Issue Security Credentials ... 225
13.5 CS02d Update Device Certificates on Device ... 230
13.6 CS02e Provide Device Certificates from Device ... 235
13.7 Pair-wise Authorisation of Devices ... 240
13.8 GCS59 / 62 GPF Device Log Backup and Restore ... 258
13.9 CS02f Provide Security Credential Details Command and Response ... 265
13.10 CS02g Update Load Controller Security Credentials Command, Response and Alert 268

14 Apply Prepayment Top Up to an ESME or GSME ... 277

14.1 Defined Terms ... 277
14.2 Description – informative ... 277
14.3 Common Requirements ... 278
14.4 CS01a Applying a Prepayment Top Up to an ESME without consumer intervention 281
14.5 CS01b Applying a Prepayment Top Up to a GSME without consumer intervention 282
14.6 Applying a Prepayment Top Up to an ESME or GSME with consumer entry of a numeric code on the ESME or GSME ... 283
14.7 Applying a Prepayment Top Up to an ESME or GSME with consumer entry of a numeric code on a PPMID ... 286
14.8 Calculating and Verifying the UTRN Check Digit ... 287

15 Message Codes ... 290

16 Event / Alert Codes and related requirements ... 291

16.1 Introduction – informative ... 291
16.2 Event and Alert Codes ... 292
16.3 Event Logs ... 292
16.4 Requirements ... 292

17 Remote Party Usage Rights ... 296

17.1 Remote Party Access Rights to Attributes and Methods ... 296
17.2 Remote Party Usage Rights to Use Cases ... 296

18 Message Templates ... 297

18.1 GBZ and ZSE Message Templates ... 297
18.2 DLMS COSEM Message Templates ... 300
18.3 Illustrative command and response instantiation and DER encoding ... 329
18.4 Cryptographic Test Vectors ... 339

19 Use Cases ... 357

19.1 Use Case Title ... 357
19.2 Use Case-specific content ... 359
19.3 Embedded Use Cases ... 360

20 Mapping Table ... 362

21 Glossary ... 363

22 Annex 1 – Additional DLMS Class ... 380

22.1 Attribute description ... 380
22.2 Method description ... 381

23 Annex 2 – This Section is intentionally left blank ... 383

24 Annex 3 – This Section is intentionally left blank ... 384

25 Annex 4 – Use of ZigBee in GBCS – informative ... 385

25.1 Purpose ... 385
25.2 GBCS requirements to use ZigBee ... 385
25.3 GBCS requirements not to use ZigBee / vary from it ... 385

26 Annex 5 – Use of DLMS COSEM in GBCS – informative ... 387

26.1 Purpose ... 387
26.2 GBCS requirements to use DLMS COSEM ... 387
26.3 GBCS requirements not to use DLMS COSEM / vary from it ... 388

27 Annex 6 – Deducing the UTRN Counter from the Truncated UTRN Counter – informative ... 389

28 Annex 7 – Data Item Values to be set prior to installation of Devices ... 391

1 Introduction – normative

The Gas Smart Metering Equipment Technical Specifications (GSMETS) requires that Gas Smart Metering Equipment (GSME), and the Electricity Smart Metering Equipment Technical Specifications (ESMETS) requires that Electricity Smart Metering Equipment (ESME) including variants, meet the requirements described in the Great Britain Companion Specification (GBCS).

ESME shall be certified by NCSC as compliant with a relevant version of the 'Commercial Product Assurance Security Characteristic Smart Metering – Electricity Smart Metering Equipment' as identified in the Smart Energy Code Section A.

GSME shall be certified by NCSC as compliant with a relevant version of the 'Commercial Product Assurance Security Characteristic Smart Metering – Gas Smart Metering Equipment' as identified in the Smart Energy Code Section A.

The Communications Hub Technical Specifications (CHTS) requires that Communications Hubs meet the requirements described in the GBCS.

A Communication Hub shall be certified by NCSC as compliant with a relevant version of the 'Commercial Product Assurance Security Characteristic Smart Metering - Communications Hub' as identified in the Smart Energy Code Section A.

The HAN Connected Auxiliary Load Control Switches (HCALCS) Technical Specifications (HCALCSTS) requires that HCALCS meet the requirements described in the GBCS.

A HAN Connected Auxiliary Load Control Switch, to meet the requirements of this version of the GBCS, shall be certified by NCSC as compliant with a relevant version of the 'Commercial Product Assurance Security Characteristic Smart Metering – HAN Connected Auxiliary Load Control Switch' as identified in the Smart Energy Code Section A.

The Prepayment Interface Device (PPMID) Technical Specifications (PPMIDTS) requires that PPMID meet the requirements described in the GBCS.

The In Home Display (IHD) Technical Specifications (IHDTS) requires that IHD meet the ZSE requirements described in the GBCS[1].

The Standalone Auxiliary Proportional Controller (SAPC) Technical Specifications (SAPCTS) requires that SAPC meet the requirements described in the GBCS.

SAPC shall be certified by NCSC as compliant with a relevant version of the 'Commercial Product Assurance Security Characteristic - Standalone Auxiliary Proportional Controller' as identified in the Smart Energy Code Section A.

The versions of GSMETS, ESMETS, PPMIDTS, HCALCSTS, SAPCTS and CHTS to which this version of GBCS is relevant are identified in the Smart Energy Code Section A.

GBCS was notified to the European Commission in accordance with the requirements of the Technical Standards and Regulations Directive[2] laying down a procedure for the provision of information in the field of technical regulations and rules on Information Society services.

[1] The only GBCS requirements for IHD are ZSE requirements.

[2] GBCS v2.0 was notified (2017/0350/UK) under Article 8 of Directive 98/34/EC of the European Parliament and of the Council (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). Directive 98/34/EC has now been replaced by Directive 2015/1535/EU of the European Parliament and of the Council (OJ L 241, 17.9.2015, p.1), which came into force on 7 October 2015

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.1[3]) specified structures defined in this document, or in IETF RFC 5912[4]. 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 'Technical Specifications' (GSMETS, ESMETS, SAPCTS IHDTS, HCALCSTS, PPMIDTS and CHTS):
  - 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) and SAPC 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 the Technical Specifications;
- the end to end protocol that is defined in the GBCS, which deviates from the standard ZigBee SEP 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 areas that would not require certification please see Section 25 for ZigBee SEP, and Section 26 for DLMS COSEM.

GB Smart Metering requires technical interoperability, and so requires a <u>single, consistent, technical</u> implementation of the capabilities laid out in the Technical Specifications 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 Single Element, Polyphase, Twin Element, Auxiliary Load Control Switch (ALCS), Auxiliary Proportional Controller (APC) and Boost Function variants thereof;

[3] http://www.itu.int/rec/T-REC-X.680-X.693-201508-I/en
[4] http://tools.ietf.org/html/rfc5912

- Gas Smart Metering Equipment (GSME);
- Communications Hub Function (CHF) and Communications Hub, Gas Proxy Function (GPF);
- Standalone Auxiliary Proportional Controller (SAPC), including Auxiliary Load Control Switch (ALCS) and Boost Function variants thereof;
- Prepayment Interface Device (PPMID) and HAN Connected Auxiliary Load Control Switch (HCALCS); and
- Type 2 Devices, including In Home Displays (IHDs).

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

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

3 Scope and Terminology

3.1 Introduction – informative

This Section 3.1 is informative and summarises Section 3.

Section 3 introduces key terms used in the GBCS:

- Messages are how Devices communicate between themselves and with organisations remote from Consumers' Premises. Such Messages are 'end-to-end' and 'unicast' in that:
  - they all identify the sender (e.g. a Supplier) and the intended recipient (e.g. an ESME); and
  - they are all intended for processing by the intended recipient, even though they may pass through intermediate Devices, such as a Communications Hub Function. Most Messages pass through Communications Hub Functions unaltered, save for any 'wrapping' information needed for transport purposes. The only exceptions are where a CHF or GPF is the intended recipient or is the sender (in these cases the Message is processed by the CHF or GPF), or where covered by the Tapping Off Mechanism (Section 10);
- Messages are one of:
  - a Command to a Device or a corresponding Response;
  - an Alert from a Device; or
  - an information provision transaction (HAN Only Message) solely between Devices;
- Organisations (such as Suppliers and Network Operators) communicating with Devices are called Remote Parties;
- Messages to and from Remote Parties are called Remote Party Messages; and
- Messages solely between Devices are called HAN Only Messages.

Section 3 then:

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

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

3.2 Scope

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

A Message shall be of one the following:

- a Command;
- a Response to a Command;
- an Alert; or

- an information provision transaction (HAN Only Message).

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

The Technical Specifications define the minimum functional capabilities required of Devices.

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

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

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

There are two classifications of Message:

- HAN Only Message[5], where both the original sender and ultimate recipient are Devices within the same Smart Metering Home Area Network (SMHAN); and
- Remote Party Message, where either the original sender or the ultimate recipient is not a Device.

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

- a Command;
- a Response to a Command; or
- an Alert.

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

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

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

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

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

3.3 Terminology

3.3.1 Numbers

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

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

[5] 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.

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

3.3.2 Bit numbering

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

3.3.3 Octets and bytes – informative

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

3.3.4 Tag and MAC – informative

In this GBCS:

- the word 'tag' is always used in the sense it is meant in encoding standards, such as A-XDR[6] and Distinguished Encoding Rules (DER)[7];
- 'tag' is never used to mean Authentication tag, in the cryptographic sense;
- 'MAC' is always used to mean Message Authentication Code, which is a cryptographic checksum on data. Thus, MAC is used instead of Authentication tag; and
- 'MAC' is never used to refer to Medium Access Control, as used in 'MAC address', which is a unique identifier assigned to network interfaces.

3.3.5 Concatenation

X || Y shall mean the concatenation of the two octet strings X and Y.

For example:

X = 0xCAFE
Y = 0xBEEF
X || Y = 0xCAFEBEEF

3.3.6 Encoding and length of variable length unsigned integers

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

- if 0<X<128, then Encoding(X) is a single octet whose value is X; or
- if 128<= X <256, then Encoding(X) is a an octet string composed of the concatenation 0x81 || Y, where Y is one octet in length and has a value equal to X; or
- if 256<= X <65,536, then Encoding(X) is a an octet string composed of the concatenation 0x82 || Y, where Y is two octets in length and has a value equal to X; or
- if 65,536<= X <16,777,216, then Encoding(X) is a an octet string composed of the concatenation 0x83 || Y, where Y is three octets in length and has a value equal to X.

Len(Encoding(X)) shall be the length in octets of Encoding(X).

3.3.7 GeneralizedTime

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

[6] IEC 61334-6
[7] https://www.itu.int/rec/T-REC-X.690/en

3.3.8 Octet Endianness – Informative

Some fields (or elements) are one octet long, for example an 8 bit integer field.

Some fields are made up of more than one octet. For example a 16 bit integer is made up of two octets. There are two ways of expressing these multiple octet elements, called 'big endian' or 'little endian'. Big endian means that the octets are listed from most significant octet first to least significant octet last. Little endian is the reverse. Whether a field is expressed as little or big endian is called endianness.

To illustrate, the decimal 500 can be represented as the hexadecimal value 0x01F4. As a 16 bit integer, its big endian representation is 0x01 then 0xF4 and its little endian representation is 0xF4 then 0x01.

Endianness does not affect single octet elements, nor does it affect the sequence of elements. Thus, it does not affect sequences of single octet elements such as 8 bit integer arrays, byte arrays, octet strings or character strings.

The GBCS defaults to 'big endian' except for parts where a protocol specifies 'little endian' sequence. The little endian requirements relate to ZigBee, specifically:

- The ZigBee OTA specification requires the OTA Header elements to be 'little endian'. GBCS requires the use of the OTA Header and so the 'little endian' requirements of ZigBee OTA apply to these fields (see Section 11.2.3); and
- The ZCL specification requires elements to be 'little endian' in ZCL commands. This applies to all ZCL and ZSE commands, including those required by the GBCS (see Section 7.4 and elsewhere in the GBCS). It also applies to the ZCL header and ZCL payload parts of GBZ Payloads, as required by Section 7.2. Note that all other elements of the GBZ Payloads, all DLMS COSEM Payloads, all ASN.1 Security Payloads and all other parts of Messages constructed according to Section 7.2 are 'big endian'. This is because they are defined in the GBCS and not in the ZCL specification. Thus, the GBZ Payload elements labelled 'Profile ID', 'Alert Code', 'Timestamp', 'Extended Header Cluster ID', 'Extended Header GBZ Command Length', 'From Date Time' and 'Length of Ciphered Information' are big endian. Note that the GBZ Payload element 'Ciphered Information' is an octet string and so endianness is irrelevant (since it is a sequence of single octet elements).

Note that GBCS does not use the two octet 'Manufacturer code' field in the ZCL Header. The other three fields in the ZCL Header are single octet elements and so endianness has no effect in relation to ZCL Header.

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 following additional points are to be noted:

- the governance and structures to ensure uniqueness of identifiers are set out in the Smart Energy Code (SEC) 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 both Encryption and a MAC), to the Supplementary Remote Party. In all other cases, the Response containing Supplementary Remote Party details is protected back to the Access Control Broker; and

- 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 support Protection Against Replay functionality.

4.2 Cryptographic Protections applying to all Messages

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

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

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

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

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

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

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

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

4.3 Security for Remote Party Messages

This Section 4.3 shall:

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

4.3.1 Identifiers, Counters and Protection Against Replay

4.3.1.1 Identifiers

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

Entity Identifiers shall be used in the Business Originator ID and Business Target ID fields of 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

Message Type	Business Originator ID	Business Target ID
		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

For clarity, Commands must contain the correct Entity Identifier value in the Business Originator ID field of Commands to align to Device Requirements. Specifically, the Business Originator ID field in a Command:

- Must contain the Entity Identifier value in the `{remotePartyRole, digitalSignature, management}` Trust Anchor Cell, if the Command requires a KRP signature;
- Must contain the Entity Identifier value in the `{supplier, keyAgreement, prePaymentTopUp}` Trust Anchor Cell, if the Command is a Prepayment Top Up; and
- Must contain the Entity Identifier value in the `{remotePartyRole, keyAgreement, management}` Trust Anchor Cell, if the Command is not a Prepayment Top Up and does not require a KRP Signature.

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

Table 4.3.1.2: Responsibility for generating the Originator Counter

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*':

- Supplementary Remote Party ID, which shall be the Entity Identifier of the Remote Party requesting the creation of the Command by the Access Control Broker; and
- Supplementary Remote Party Counter, which shall be an octet string of length 8.

All Responses to such Commands shall incorporate:

- the same Supplementary Remote Party ID and Supplementary Remote Party Counter as the Command; and
- for those marked as '*Supplementary Originator Counter required in Response*' in the 'Use Case reference' worksheet of the Mapping Table', a Supplementary Originator Counter which shall be generated by the Device, and shall be an octet string of length 8 whose contents shall be set and read as an unsigned 64-bit integer. 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.

4.3.1.5 Protection Against Replay mechanisms

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

For each type of Command that a Device supports, and that is marked as 'Protection Against Replay Required' in the Use Case Reference tab of the Mapping Table, the Device shall:

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

For clarity, an SAPC is not required to have the capability to store an Execution Counter for a Message Code where the corresponding Use Case:

- is marked as 'optional' in the 'SAPC support' column of the 'Use Case reference tab' in Table 20; and

- is not supported by the SAPC.

4.3.2 Security Credentials

4.3.2.1 Introduction – informative

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

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

Sections 8 and 13 cover the above functionality.

Section 12 covers requirements related to the structure and content of such Security Credential Documents, where such requirements are relevant to Device processing requirements.

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

4.3.2.2 Security Credential Documents

A Security Credential Document shall be either:

- a Device Certificate; or
- a Remote Party's Organisation Certificate; or
- a Certification Authority Certificate.

4.3.2.2.1 Device Certificate

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

4.3.2.2.2 Remote Party's Certificate

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

Security Credentials shall be incorporated in the Message in the form of the Remote Party's Certificate.

4.3.2.2.3 Certification Authority Certificate

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

4.3.2.3 Device Security Credentials

Where a Device is of `deviceType` that is `gSME`, `eSME`[8], `communicationsHubCommunicationsHubFunction`, or `communicationsHubGasProxyFunction`, that Device shall have the capacity to store and use securely four private keys:

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

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

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

These stores shall be referred to as Private Key Cells.

Wherever one of a Device's Private Keys is required to be used by a GBCS Cryptographic Protection process, only the relevant Current Private Key shall be used. A Device shall not use any Pending Private Key in any GBCS Cryptographic Protection.

Where a Device holds a Private Key that is to be used for Key Agreement, the corresponding Public-Private Key Pair shall have been generated according to the NSA's '*Suite B Implementer's Guide to FIPS 186-3 (ECDSA), February 3, 2010*'[9] using the 'ECC Key Pair Generation Using Extra Random Bits' method.

Where a Device holds a Private Key that is to be used for Digital Signing, the corresponding Key Pair shall have been generated according to the NSA's '*Suite B Implementer's Guide to FIPS 186-3 (ECDSA), February 3, 2010*'[10] using the 'ECC Key Pair Generation Using Extra Random Bits' method.

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

- have two Trust Anchor Cells to store two Device Certificates relating to itself, with one Trust Anchor Cell for storing Device Certificates where `keyUsage = keyAgreement` and one for Device Certificates where `keyUsage = digitalSignature`;

[8] So including SAPC as per Section 4.3.2.5.
[9] https://www.iad.gov/iad/library/ia-guidance/ia-solutions-for-classified/algorithm-guidance/suite-b-implementers-guide-to-fips-186-3-ecdsa.cfm
[10] https://www.iad.gov/iad/library/ia-guidance/ia-solutions-for-classified/algorithm-guidance/suite-b-implementers-guide-to-fips-186-3-ecdsa.cfm

- where those two Trust Anchor Cells are populated, ensure the Device Certificates have the following attributes:
  - both Device Certificates meet the requirements specified at Section 13;
  - both Device Certificates' `hwSerialNum` fields have a value the same as the Devices' Entity Identifier; and
  - each Device Certificate's `keyUsage` field has the same value as the Trust Anchor Cell in which it is placed.

4.3.2.4 Remote Party Security Credentials

A Device shall only action a Remote Party Command where:

- the Known Remote Party identified by the Command has, according to the Security Credentials held on the Device, a Remote Party Role which, according to the Mapping Table for the Message Code in question, is allowed to request execution of the Command; and
- the Cryptographic Protections in the Command instance received by the Device have been verified, in line with the requirements for a Command with the Message Code in question.

To enable this, Security Credentials relating to the Remote Parties in question:

- shall be held in Trust Anchor Cells on the Device; and
- shall act as the corresponding Trust Anchors.

4.3.2.5 Required Trust Anchor Cells and related Device requirements

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

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

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

```
DeviceType ::= INTEGER {
   gSME                                          (0),
   eSME                                          (1),
   communicationsHubCommunicationsHubFunction    (2),
   CommunicationsHubGasProxyFunction             (3),
   type1HANConnectedAuxiliaryLoadControlSwitch   (4),
   type1PrepaymentInterfaceDevice                (5),
   type2                                         (6)
}
```

An SAPC shall have a `deviceType` of `eSME (1)` throughout this GBCS.

Every Device shall:

- have storage allocated capable of holding Security Credentials as required by Table 4.3.2.5 for its Device type; and
- 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 /SAPC	GSME	CH (CHF)	CH (GPF[11])	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	✓ [12]	✓				
6	networkOperator	digitalSignature	management	✓			✓		
7	networkOperator	keyAgreement	management	✓			✓		
8	accessControlBroker	digitalSignature	management			✓			✓
9	accessControlBroker	keyAgreement	management	✓	✓	✓	✓	✓	✓
10	transitionalCoS	digitalSignature	management	✓	✓		✓	✓	
11	wanProvider	digitalSignature	management			✓			
12	loadController	digitalSignature	management	✓					
13	loadController	keyAgreement	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;

[11] Supplier and Network Operator credentials on the Communications Hub (Gas Proxy Function) 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 Function functionality, apart from Update Security Credentials, is inoperable)

[12] This Trust Anchor Cell is required on an SAPC, recognising that, in line with SMETS, the SAPC may or may not support prepayment related functionality.

- 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 Commands required by Section 13; and
- shall not allow execution of any Remote Party Command other than an Update Security Credentials Command or 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 encrypting data 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

TrustAnchorCellIdentifier			Usage of the Public Key in the Trust Anchor Cell
remotePartyRole	keyUsage	cellUsage	
			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 (CHF)
loadController	digitalSignature	management	Used to check the signature of the loadController on Critical Commands the loadController has sent to the Device
loadController	keyAgreement	management	Used in encrypting data in Alerts and Responses addressed to the loadController

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

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

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

4.3.2.7.1 Message Authentication Codes

Where a Command is a Prepayment Top Up Command, the `supplier` MAC in that Command shall be verified using the Public Key in Trust Anchor Cell {**remotePartyRole** *supplier*, **keyUsage** *keyAgreement*, **cellUsage** *prePaymentTopUp*}, along with the Device's Key Agreement Private Key.

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

4.3.2.7.2 Signature

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

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

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

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

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

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

1. supplier;
2. wanProvider;
3. networkOperator;
4. accessControlBroker.

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

4.3.2.8 Certification Path Validation

4.3.2.8.1 Access Control Broker requirements

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

- either by using the algorithm specified in IETF RFC 5280[13] section 6.3; or
- by using functionality equivalent to the external behaviour resulting from that algorithm.

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

4.3.2.8.2 Device requirements

The requirements in this Section 4.3.2.8.2 shall apply only to Use Cases CS02b (Update Security Credentials) and CS02g (Update Load Controller Security Credentials).

[13] http://datatracker.ietf.org/doc/rfc5280/

Where a Device has successfully completed all required Command Authenticity and Integrity checks on a Command, of a type covered by Use Cases CS02b and CS02g, the Device shall undertake either:

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

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

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

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

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

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

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

If the Device's Certificate Path Validation does not confirm the required certification path validity, then the Device shall undertake no further processing of the Command, except for the issuance of a Message containing an `executionOutcome` (with its Section 13.3.4.6 meaning) notifying that the Command was unsuccessful in applying at least some of the changes in the Command.

4.3.2.9 DLMS Client and Server

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

4.3.3 Cryptographic primitives and their usage

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

- use SHA-256, as specified in *FIPS 180-4*[14], as the Hash function;
- use the AES-128 cipher, as specified in *FIPS 197*[15], as the block cipher primitive;
- use the Galois Counter Mode (GCM) mode of operation as specified in *NIST Special Publication 800-38D*[16] ;
- use the GMAC technique, based on the use of AES-128, for the calculation of Message Authentication Codes (MACs), as specified in *NIST Special Publication 800-38D* (see above);

[14] http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf
[15] http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
[16] http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf

- use, as the Digital Signature technique, ECDSA (as specified in *FIPS PUB 186-4*) in combination with the curve P-256 (as specified in *FIPS PUB 186-4* at section D1.2.3) and SHA-256 as the Hash function. Within Messages, Signatures shall be in the Plain Format;
- use, to calculate the Shared Secret Z, the Static Unified Model, C(0e, 2s, ECC CDH) Key Agreement technique (as specified in *NIST Special Publication 800-56Ar2*[17] save for the requirement to zeroize the Shared Secret) with:
  - the Single-step Key Derivation Function (KDF) based on SHA-256, as specified in *NIST Special Publication 800-56Ar2*; and
  - the P-256 curve for the elliptic curve operations.

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

4.3.3.1 Scope of Cryptographic Protections

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

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

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

4.3.3.2 ECDSA per message secret number

When generating a Digital Signature, the Smart Metering Entity shall calculate the DSA Per-Message Secret Number 'k' with respect to ECDSA (with the meaning in section 6.3 of *FIPS 186-4*) to be the SHA-256 hash of the concatenation of:

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

If the value of k so calculated is zero or greater than n -1, or results in an 'r' or 's' value of 0, where r and s have the meanings in the NSA's '*Suite B Implementer's Guide to FIPS 186-3 (ECDSA)*', then a new value for k shall be calculated to be the SHA-256 hash of the concatenation of:

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

The addition of 0x00 to the concatenation shall be repeated until a value of k is generated that does not result in k being zero or greater than n -1, or an 'r' or 's' value of 0.

[17] http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar2.pdf

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

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

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

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

where:

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

4.3.3.4 Calculating the Initialization Vector for GCM and GMAC

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

FixedField || *InvocationField*

where:

- *FixedField* shall always have the same value as the Business Originator ID in the Grouping Header part of the Message being processed (see Section 7.2.7); and
- *InvocationField* = 0x00000000.

The DLMS COSEM Authentication Key (AK), as defined in the Green Book, shall be a zero length string.

4.3.3.4.1 Other input parameters to MAC and Encryption / Decryption operations – informative

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

The template for such tables is the Table 4.3.3.4.1. Please note that this table does not 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:		

Input Parameter	Value	Note
Additional Authenticated Data shall be the concatenation:		

Table 4.3.3.4.1: Template for other input parameters

4.3.3.4.2 Size of MAC

The bit length of the MAC shall be 96 except for the MAC contained in the WrappedApexContingencyKey extension within root Certificates, where the bit length of the MAC shall be 128.

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.

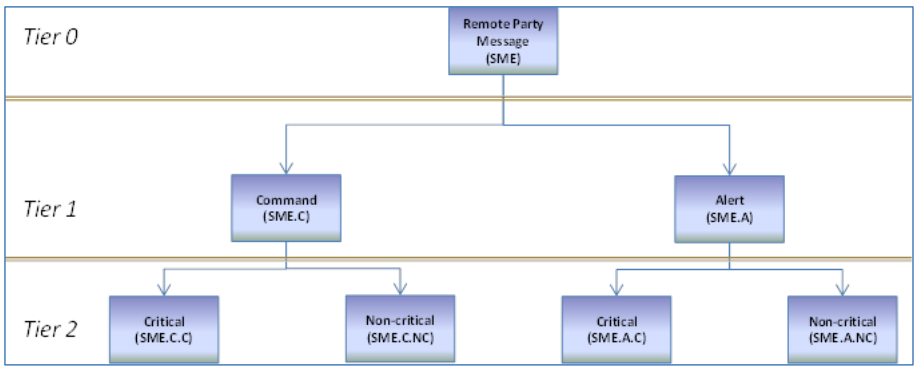

Figure 5.3: Message Categories

Note that the 'Command' part of the hierarchy covers requirements for both the Command and the corresponding Response. Except in certain error cases (e.g. cryptographic processing failure), a Command always leads to a Response.

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

5.4 Common Message Processing steps – informative

A common set of stages for Remote Party Message processing is used in this GBCS and the Use Cases, except for Variant Messages[18]. Variant Messages include Security Credentials and Prepayment Top Up related Messages.

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
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 Authenticity 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 verifying the Access Control Broker's MAC	Device

Table 5.4a: Common stages for Commands

[18] See Mapping Table for identification of Variant Messages

That common set of stages for Responses is shown in Table 5.4b.

Name of Stage	Summary of the stage	Responsible Smart Metering Entity
i. Response Construction	The Response is fully populated by the Device, apart from cryptographic fields	Device
ii. Response Cryptographic Protection	The Device adds the required Cryptographic Protection to the Response	Device
iii. 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

Table 5.4b: Common stages for Responses

That common set of stages for Alerts is shown in Table 5.4c.

Name of Stage	Summary of the stage	Responsible Smart Metering Entity
i. Alert Construction	The Alert is fully populated by the Device, apart from cryptographic fields	Device
ii. Alert Cryptographic Protection	The Device adds the required cryptographic fields to the Alert	Device
iii. 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

Table 5.4c: Common stages for Alerts

The generic processing applied to Commands and their Responses (in relation to integrity, authenticity and non-repudiation) in a Message Category 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:* Verifies digital signature	↓	*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

Table 5.4d: Generic Command and Response processing

The generic processing applied to Alerts in a Message Category is summarised in Table 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

Table 5.4e: Generic Alert processing

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.

For clarity, this Section 6 does not detail requirements for ZigBee commands or messages that may result from a Device processing Remote Party Messages.

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
i. Command Construction	The entity undertaking this phase is not known to the Device
ii. Command Cryptographic Protection I	Known Remote Party
iii. Command Cryptographic Protection II	Access Control Broker / PPMID
iv. 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
i. Response Construction	Device

Stage	Responsible Smart Metering Entity
ii. Response Cryptographic Protection	Device
iii. 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: 0x110000000000 || Grouping Header || Command Payload|| 0x40 || KRP Signature Where a KRP Signature is not present: 0x110000000000 || 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.

For HAN Only Commands from the PPMID to a GSME, the PPMID shall calculate the PPMID to GSME Message Authentication Code (PPMID-GSME MAC) using the parameters 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:	0x110000000000 || Grouping Header || Command Payload || 0x00	

Table 6.2.3b: Calculation of PPMID-GSME MAC

The PPMID-GSME 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.

6.2.4 Command Authenticity and Integrity Verification

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

6.2.4.1 Checks to be undertaken

The Device shall undertake the checks in Section 6.2.4.1.1 before any other checks in this Section 6.2.4.1, and shall undertake the other checks in the sequence set out in this Section 6.2.4.1, except, where relevant, as specified in Sections 13.5.4 and 13.7.4.2.2, before undertaking any other processing of the Command.

6.2.4.1.1 Message Identifier Validation

The Device shall verify that:

1. the Business Target ID in the Command has the same value as the Device's Entity Identifier;
2. the Message Code is for a Message that the Device is capable of processing, according to the associated Use Case. For an SAPC:
    a. this test shall be applied as if the SAPC were Single Element Electricity Metering Equipment (with its SMETS meaning), and so ESME as defined in Part A of Section 5 of SMETS;
    b. this test shall succeed if the Use Case states that an ESME is required to support it, regardless of whether the SAPC does support the corresponding Use Case; and
    c. where the SAPC does not support the corresponding Use Case, the SAPC shall not undertake any of the subsequent checks in this Section 6.2.4.1.1;

3. 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); and
4. the contents of the Message Payload conform to the GBCS in that:
   - where the Use Case, as identified by its Message Code, specifies a DLMS COSEM Payload:
     - the DLMS COSEM Payload complies with the Green Book section 9.5 ASN.1 definition of *@CosemPDU.XDLMS-PDU.access-request*; and
     - the presence and order of *@CosemPDU.XDLMS-PDU.access-request.access-request-body.access-request-specification* is as per the corresponding Message Template, excluding the order of content within any occurrences of the *Selective-Access-Descriptor* structure; and
     - all parameters within the *@CosemPDU.XDLMS-PDU.access-request* that identify a script table or a script within a script table are within the range of values specified in the corresponding Use Case;
   - where the Use Case, as identified by its Message Code, specifies a GBZ Payload:
     - the combination of values in the Extended Header Cluster ID and ZCL Header Command ID fields in each GBZ Use Case Specific Component are explicitly required in the corresponding Message Template; and
     - for Critical Commands, the GBZ Use Case Specific Component, as identified by Extended Header Cluster ID and ZCL Header Command ID fields, are in the order defined in the corresponding Message Template;
   - where the Use Case, as identified by Message Code, specifies an ASN.1 Security Payload, the Payload conforms to the requirements of the corresponding Use Case.

6.2.4.1.2 ACB-SMD MAC Verification

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

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:	

Input Parameter	Value	Note
	0x110000000000 || Grouping Header || Command Payload || 0x40 || KRP Signature Where a KRP Signature is not present: 0x110000000000 || 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:	0x110000000000 || 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 Device is an SAPC; and
- all of the checks required in Section 6.2.4.1 have succeeded; and
- the Device does not support processing of Commands with the Message Code in this Command

then the Device shall:

- generate an entry in the Event Log with Alert / Event Code of 0x8F85;
- discard the Command without execution and without sending a Response; and
- send a 'Command not supported by Device' Alert, constructed as required by Section 7.2.9.1.

If the Message requires 'Protection Against Replay' according to the corresponding Use Case, the Device shall, unless it is an SAPC that does not support processing of Commands with the Message Code in this Command, 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:

- generate an entry in the Security Log recording failed Authentication;
- discard the Command without execution and without sending a Response; and
- send an Alert notifying the failed Authentication, constructed as specified in Section 6.7, populated with the relevant Alert Code from Section 16 (the one of 0x8F1E, 0x8F30 or 0x8F3D that is required), to the Known Remote Party specified in Section 16. If the Device is an ESME or a CHF, the Alert Payload shall be a DLMS COSEM Alert Payload. Otherwise, the Alert Payload shall be a GBZ Alert Payload.

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

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

6.3 Message Category SME.C.C

6.3.1 Definitions

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

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

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

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

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

- Digital Signing of the Command by the Known Remote Party;
- verification of the Digital Signature in the Command by the Device;
- Digital Signing of the Response by the Device; and
- 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

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

6.3.2.2 Processing stages defined in subordinate categories

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

- Command Construction; and
- 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.

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

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

6.4.2 Processing stages

6.4.2.1 Processing stages defined in the superordinate Message Category

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

6.4.2.2 Processing stages defined in subordinate categories

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

- Command Construction; and
- Response Construction.

6.4.3 Command Cryptographic Protection I

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

6.4.4 Command Authenticity and Integrity Verification

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

6.4.5 Response Cryptographic Protection

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

The Device shall calculate the Device to Known Remote Party MAC (SMD-KRP MAC) using 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.		

Input Parameter	Value	Note
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:	0x110000000000 || Grouping Header || Response Payload || 0x00	

Table 6.4.5: Calculation of Device to Known Remote Party MAC

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

6.4.6 Response Recipient Verification

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

The Remote Party, as identified by the Business Target ID in the Response, may validate the SMD-KRP MAC in the Response by calculating a MAC using the parameters in Table 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 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 4.3.3.4, the Plaintext shall be empty and:		
Additional Authenticated Data shall be the concatenation:	0x110000000000 || Grouping Header || Response Payload || 0x00	

Table 6.4.6: MAC calculation for SMD-KRP MAC validation

6.5 Message Category SME.C.PPMID-GSME

6.5.1 Definitions

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.

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

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

6.5.2.2 Processing stages defined in subordinate categories

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

- Command Construction; and
- Response Construction.

6.5.3 Command Cryptographic Protection I

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

6.5.4 Command Authenticity and Integrity Verification

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

6.5.5 Response Cryptographic Protection

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

The GSME shall calculate the GSME to PPMID MAC (GSME-PPMID MAC) using the 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:	0x110000000000 || Grouping Header || Response Payload || 0x00	

Table 6.5.5: Calculation of GSME-PPMID MAC

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

6.5.6 Response Recipient Verification

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

The PPMID, as identified by the Business Target ID in the Response, shall validate the GSME-PPMID MAC in the Response by calculating a MAC using the parameters in Table 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:		
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 :	0x110000000000 || Grouping Header || Response Payload || 0x00	

Table 6.5.6: MAC calculation for GSME-PPMID MAC validation

6.6 Message Category SME.A

6.6.1 Definitions

The superordinate Message Category for SME.A is SME.

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

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

6.6.2 Processing Stages

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

Stage	Responsible Smart Metering Entity
i. Alert Construction	Device
ii. Alert Cryptographic Protection	Device
iii. Alert Recipient Verification	Remote Party named in the Alert

Table 6.6.2: SME.A Processing Stages

6.6.2.1 Processing stages defined in the superordinate Message Category

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

6.6.2.2 Processing stages defined in subordinate categories

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

- Alert Construction;
- Alert Cryptographic Protection; and
- Alert Recipient Verification.

6.7 Message Category SME.A.C

6.7.1 Definitions

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

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

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

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

6.7.2 Processing stages

6.7.2.1 Processing stages defined in the superordinate Message Category

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

6.7.2.2 Processing stages defined in subordinate categories

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

6.7.3 Alert Cryptographic Protection

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

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

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

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

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

6.7.4 Alert Recipient Verification

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

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

6.8 Message Category SME.A.NC

6.8.1 Definitions

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

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

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

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

6.8.2 Processing stages

6.8.2.1 Processing stages defined in the superordinate Message Category

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

6.8.2.2 Processing stages defined in subordinate categories

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

6.8.3 Alert Cryptographic Protection

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

The Device shall calculate the Device to Known Remote Party MAC (SMD-KRP MAC) using 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 Target 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:	0x110000000000 || Grouping Header || Alert Payload || 0x00	

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

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

As stated in Table 6.8.3, the Public Key Agreement Key used must be the 'Known Remote Party's' and must be that 'As held by the Device in the relevant Trust Anchor Cell `{remotePartyRole, keyAgreement, management}`'.

6.8.4 Alert Recipient Verification

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

The Remote Party, as identified by the Business Target ID in the Alert, may validate the SMD-KRP MAC in the Alert by calculating a MAC using the parameters in Table 6.8.4 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 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 4.3.3.4, the Plaintext shall be empty and:		
Additional Authenticated Data shall be the concatenation:	0x110000000000 || 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 ZigBee 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 ZigBee 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 DLMS COSEM 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).

The Technical Specifications 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 and the SAPC;
- 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 1700[19]. Elements detailed in this Section 7 as being defined in the ZSE or ZCL

[19] http://tools.ietf.org/html/rfc1700

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

7.2.1 Commands

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

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

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

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

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

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

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

7.2.2 Responses

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

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

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

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

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

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

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

7.2.3 Alerts

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

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

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

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

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

7.2.4 Payload sequence and Break On Error

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

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

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

Command, except where the instruction in the Command is a ZSE *GetDayProfiles* or *GetWeekProfiles* command where one or more ZSE commands may be produced in the Response in line with the ZSE specification. The corresponding result in the Response Payload shall detail that instruction's success or failure. The Response Payload shall explicitly detail a result for each of the subsequent instructions that were not attempted. The results in the Response shall be in the same order as the instructions in the Command.

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

- for DLMS instructions, a `Data-Access-Result` / `Action-Result` of `other-reason`;
- for each such ZCL / ZSE instruction, a Default Response command where the ZCL payload status field has a value of FAILURE (0x01), ZCL payload *Command ID* is set to 0xFF and the ZCL header Frame Control field is set to 0x00 or 0x08, with the Direction Sub-field being 0b0 or 0b1 in line with the ZigBee specifications.

A ZSE command returning a status of 'NOT_FOUND' shall not be treated as a failure.

7.2.5 Message construction – MAC Header

The required components of the MAC Header shall be populated with the values as per Table 7.2.5.[20]

MAC Header				
No	**DLMS COSEM Message Elements**	**Contents**	**Length (octets)**	**Note**
	General-Ciphering	0xDD	1	DLMS COSEM 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			

[20] See Green Book.

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

Table 7.2.5: Required components of MAC Header

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

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

The Green Book requires that the AAD used as input to the MAC calculation is the concatenation of:

SC II AK II transaction-id II originator-system-title II recipient-system-title II date-time II other-information II information to be protected

The Green Book also requires that, for the elements contributing to AAD, the lengths of certain octet string fields are included, as well as the values of all fields. The Green Book defines octet strings within the general-ciphering service where lengths have to be included as:

transaction-id *OCTET STRING,*
originator-system-title *OCTET STRING,*
recipient-system-title *OCTET STRING,*
date-time *OCTET STRING,*
other-information *OCTET STRING,*

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

- SC takes the value 0x11; and
- the following octet strings in the general-ciphering service shall have zero length (so a length octet of 0x00) and so have no value:
  - transaction-id,

- originator-system-title,
- recipient-system-title,
- date-time,
- other-information.

As required by Section 4.3.3.4, AK is a zero length octet string.

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

0x110000000000 II information to be protected

7.2.7 Message construction – Grouping Header

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

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

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

CRA Flag || Originator Counter || Business Originator ID || Business Target ID || date-time (if present) || Message Code || Supplementary Remote Party ID (if present) || Supplementary Remote Party Counter (if present) II Supplementary Originator Counter (if present) || Supplementary Remote Party Key Agreement Certificate (if present) || (information to be protected)

where (information to be protected) shall be:

- the Command Payload in a Command;
- the Response Payload in a Response; or
- the Alert Payload in an Alert.

Grouping Header				
Input to the ECDSA calculation	**DLMS COSEM Message Elements**	**Contents**	**Length (octets)**	**Note**
No	General-Signing	0xDF	1	DLMS COSEM APDU tag for General-Signing (223 in decimal)
	transaction-id			
Yes	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			
Yes	length	0x08	1	Length of Entity Identifier

Grouping Header				
Input to the ECDSA calculation	**DLMS COSEM Message Elements**	**Contents**	**Length (octets)**	**Note**
Yes	value	Business Originator ID	8	
	recipient-system-title			
Yes	length	0x08	1	Length of Entity Identifier
Yes	value	Business Target ID	8	
	date-time			
Yes	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			
Yes	length	Encoding(X)	variable Len(Encoding(X))	X is length of other information octet string. X is variable
Yes	value	Message Code || Supplementary Remote Party ID || Supplementary Remote Party Counter || Supplementary Originator Counter || Supplementary Remote Party Key Agreement Certificate	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

Grouping Header				
Input to the ECDSA calculation	**DLMS COSEM Message Elements**	**Contents**	**Length (octets)**	**Note**
				in line with the 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			
Yes	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

7.2.8 Message construction – ASN.1 Security Payloads

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

7.2.9 Message construction – DLMS COSEM Payloads

For Messages containing DLMS COSEM payloads (as defined by the Message Code and Use Cases in this GBCS):

- any Command Payload shall comply with the requirements of Table 7.2.9a and the associated Use Case;
- any Response Payload shall comply with the requirements of Table 7.2.9b and the associated Use Case; and
- any Alert Payload shall comply with the requirements of Table 7.2.9c and the associated Use Case.

DLMS COSEM Payloads – Commands				
No	**DLMS COSEM Message Elements**	**Contents**	**Length (octets)**	**Note**
	access-request	0xD9	1	DLMS COSEM 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*	*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*
2	*Use Case Specific Content*	*Use Case specific*		*The list of Gets, Sets and Actions specific to the Use Case. This content is specified in each DLMS COSEM Use Case*
	access-request-list-of-data			
	list-of-data			
3	*SEQUENCE OF Data*	*Use Case specific*	*1*	*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*

DLMS COSEM Payloads – Commands				
No	**DLMS COSEM Message Elements**	**Contents**	**Length (octets)**	**Note**
4	*Use Case Specific Content*	*Use Case specific*	*Use Case set*	*Values of the attributes required by the Use Case. This content is specified in each DLMS COSEM Use Case*

Table 7.2.9a: Required components of Command Payload

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

DLMS COSEM Payloads – Responses				
No	**DLMS COSEM Message Elements**	**Contents**	**Length (octets)**	**Note**
	access-response	0xDA	1	DLMS COSEM 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*	*The total number of attributes in the Response in the Use Case. This content is specified in each DLMS COSEM Use Case*
6	*Use Case Specific Content*	*Use Case specific*	*Use Case set*	*Values of the attributes required by the Use Case. This content is specified in each DLMS COSEM Use Case*
	access-response-specification			
7	*SEQUENCE OF CHOICE*	*Use Case specific*	*1*	*The total number of responses, including the 1 here and those in the Use Case*
8	*Use Case Specific Content*	*Use Case specific*	*Use Case set*	*Fields stating the result of each Gets, Sets and Actions specific to the Use Case.*

Table 7.2.9b: Required components of Response Payload

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

DLMS COSEM Payloads – Alerts				
No	**DLMS COSEM Message Elements**	**Contents**	**Length (octets)**	**Note**
	data-notification	0x0F	1	DLMS COSEM APDU tag for data-notification (15 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
	notification-body			
	structure	0x02	1	
1	*SEQUENCE OF Data*	*0x02 unless there is Use Case specific data additional*	*1*	*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).* *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*
	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*	*See Note at row 1, which means that, for most Alerts, there will be no Use Case specific content.*

Table 7.2.9c: Required components of Alert Payload

Elements marked in Table 7.2.9c as Use Case specific shall be populated according to the Use Case for the Message Code.

7.2.9.1 DLMS COSEM Alert population

Devices shall construct Alerts with the following Message Codes according to the requirements of this Section 7.2.9.1:

Use Case Name	ECS100 Command not supported by Device
Facet	*Value*
Grouping	Remote Party Message
Message Type	Alert
Summary	The SAPC shall create and send this Alert when it receives a Command which, in line with SMETS 9.1, it does not support but which would be supported if the SAPC were an ESME
Message Type Category	SME.A.C (see Section 6.7)
Payload	DLMS COSEM (see Table 7.2.9c)
Grouping Header Fields:	
• Message Code	0x0120
• Business Target ID	The Business Originator ID from the corresponding field in the Command
• Supplementary Remote Party ID	This field shall be included in the Alert if Supplementary Remote Party ID is present in the Command, and it shall take the same value as in the Command
• Supplementary Remote Party Counter	This field shall be included in the Alert if Supplementary Remote Party Counter is present in the Command, and it shall take the same value as in the Command
Payload Fields:	
• Alert Code	0x8F85
• Use Case Specific Additional Content	The concatenation: 0x09 || 0x0A || Message Code || Originator Counter where the last two fields have the values from the corresponding part of the Command. Note that 0x09 is the DLMS COSEM tag for octet-string and 0x0A is the length of the concatenation Message Code || Originator Counter

Table 7.2.9.1a: Use Case - ECS100 Command not supported by Device

Use Case Name	ECS101 Limit APC [n] Level Command processed
Facet	*Value*
Grouping	Remote Party Message
Message Type	Alert
SMETS Reference(s)	The Device shall create and send this Alert as required by the SMETS sections detailed in Table 16.2 for this Alert Code
Message Type Category	SME.A.C (see Section 6.7)
Payload	DLMS COSEM (see Table 7.2.9c)
Grouping Header Fields:	
• Message Code	0x0121
• Business Target ID	The Supplier's Entity Identifier
• Supplementary Remote Party ID	This field shall be absent

Payload Fields:	
• Alert Code	0x8F86
• Use Case Specific Additional Content	See the Message Template for 'ECS101 Limit APC [n] Level Command processed' in Section 18.2

Table 7.2.9.1b: Use Case - ECS101 Limit APC [n] Level Command processed

Use Case Name	ECS102 Limit APC [n] Level ended or cancelled
Facet	*Value*
Grouping	Remote Party Message
Message Type	Alert
SMETS Reference(s)	The Device shall create and send this Alert as required by the SMETS sections detailed in Table 16.2 for this Alert Code
Message Type Category	SME.A.C (see Section 6.7).
Payload	DLMS COSEM (see Table 7.2.9c)
Grouping Header Fields:	
• Message Code	0x0122
• Business Target ID	The Supplier's Entity Identifier
• Supplementary Remote Party ID	This field shall be absent
Payload Fields:	
• Alert Code	0x8F87
• Use Case Specific Additional Content	See the Message Template for 'ECS102 Limit APC [n] Level ended or cancelled' in Section 18.2

Table 7.2.9.1c: Use Case - ECS102 Limit APC [n] Level ended or cancelled

Use Case Name	ECS200 Operational Update
Facet	*Value*
Grouping	Remote Party Message
Message Type	Alert
SMETS Reference(s)	The Device may create and send this Alert when it wishes to notify either the Supplier, Network Operator or Load Controller (or any two of them) of some change in operational status. For clarity, GBCS does not specifiy the value of the utf8-string in 'Use Case Specific Additional Content'. However, the utf8-string shall be limited in length so that the total size of the Alert created does not exceed the Maximum PDU Size
Message Type Category	SME.A.C (see Section 6.7)
Payload	DLMS COSEM (see Table 7.2.9c)
Grouping Header Fields:	
• Message Code	0x0123
• Business Target ID	The Entity Identifier of either the Supplier, Network Operator or Load Controller
• Supplementary Remote Party ID	If included, the Entity Identifier of either the Supplier, Network Operator or Load Controller

Payload Fields:	
• Alert Code	0x8F88
• Use Case Specific Additional Content	The concatenation: 0x0C || Encoding (Len('value')) || 'value' where 'value' is the content of the UTF-8 string the Device wishes to send, which shall comply with Section 7.2.9.2. Note that 0x0C is the DLMS COSEM tag for utf8-string

Table 7.2.9.1d: Use Case – ECS200 Operational Update

7.2.9.2 JavaScript Object Notation (JSON) in DLMS COSEM utf8-strings

Where a Use Case specifies that the value of a DLMS COSEM utf8-string must comply with this Section 7.2.9.2, then the value of the utf8-string shall:

- comply with the requirements of IETF RFC8259[21];
- where it is part of a Remote Party Response, be limited in length so that the Response does not exceed 63 times 1,149 octets, so that the Response can be sent in at most 63 GBT Messages;
- where it is part of a Remote Party Alert, be limited in length so that the Alert does not exceed Maximum PDU Size octets;
- where it is to indicate that the Device does not hold information, be the string:

  {}

  This includes the case where the Device is required to report information for Auxiliary Controller [n], but there is currently no Auxiliary Controller associated with Auxiliary Controller [n], where those terms have their SMETS meaning; and
- where the Device is reporting Auxiliary Controller [n] State and there is currently an Auxiliary Controller associated with Auxiliary Controller [n], be the string:

  {"outputState": *p,* "inputState": *q other-data* }

  where:
  - *p* is an integer value between 0 and 100 inclusive specifying the Auxiliary Controller [n]'s output state (which shall correspond to the commanded output state value in Section 7.3.6.1);
  - *q* is an integer value between 0 and 100 inclusive specifying the Auxiliary Controller [n]'s input state (which shall correspond to the commanded input state value in Section 7.3.6.1); and
  - *other-data* is any other UTF-8[22] encoded information the Device includes, subject to the value of the whole string conforming to IETF RFC8259. For clarity, *other-data* may be the empty string.

7.2.10 Message construction – GBZ Payloads

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

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

[21] https://tools.ietf.org/html/rfc8259
[22] https://tools.ietf.org/html/rfc3629

- any Command Payload shall comply with the requirements of Table 7.2.10a and the associated Use Case;
- any Response Payload shall comply with the requirements of Table 7.2.10b and the associated Use Case; and
- any Alert Payload shall comply with the requirements of Table 7.2.10c and the associated Use Case.

Each GBZ Use Case Specific Component shall comply with:

- Table 7.2.10d if the ZSE / ZCL command within it is not encrypted; or
- 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	*Total number of GBZ Use Case Specific Component(s)*	*See 'Note' column*	*1*	*This octet is to be interpreted as an 8 bit unsigned integer specifying the total number of GBZ Use Case Specific Component(s)*
2	*GBZ Use Case Specific Component(s)*	*Use Case specific*		*See Tables 7.2.10d and 7.2.10e*

Table 7.2.10a: Required components of GBZ Command Payload

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

GBZ Payloads – Response				
No	**Message Elements**	**Contents**	**Length (octets)**	**Note**
	Profile ID	0x0109	2	ZSE
1	*Total number of GBZ Use Case Specific Component(s)*	*See 'Note' column*	*1*	*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*
2	*GBZ Use Case Specific Component(s)*	*Use Case specific*		*See Tables 7.2.10d and 7.2.10e*

Table 7.2.10b: Required components of GBZ Response Payload

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

GBZ Payloads – Alerts				
No	**Message Elements**	**Contents**	**Length (octets)**	**Note**
	Profile ID	0x0109	2	ZSE
1	*Total number of GBZ Use Case Specific Component(s)*	*See 'Note' column*	*1*	*0x00 if there is no Use Case Specific Component or 0x01 if there is*

GBZ Payloads – Alerts				
No	**Message Elements**	**Contents**	**Length (octets)**	**Note**
	Alert Code	See 'Note' column	2	The Alert Code for this Alert as defined in Section 16
	Timestamp	UTCTime	4	The *UTCTime*, with its ZCL meaning, at which this Alert was created
2	*GBZ Use Case Specific Component(s)*	*Use Case specific*		*For Alert Code 0x8F0A, see GCS53 and Table 7.2.10e. For Alert Codes 0x8F1C and 0x8F72, see Section 11.2.6. For Alert Codes 0x8F66 and 0x8F67, see Section 9.2.2.6. For Alert Code 0x81A0, see Section 16.4*

Table 7.2.10c: Required components of GBZ Alert Payload

Elements marked in Table 7.2.10c as Use Case specific shall be populated according to the 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, 0x11 or 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 in 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

GBZ Use Case Specific Component without encrypted content				
No	**Message Elements**	**Contents**	**Length (octets)**	**Note**
				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, except where the GBCS specifies otherwise

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

7.2.11 Transfer of Large Remote Party Messages[23]

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.

7.2.11.1 GBT Terminology and Parameters

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

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.

For a Remote Party Command sent using a GBT Message Series, the GBT Initiator shall be the Access Control Broker, or the HHT for Remote Party Commands delivered via the HHT and the GBT Recipient shall be the target Device.

For a Remote Party Response or a Remote Party Alert sent using a GBT Message Series, the GBT Initiator shall be the sending Device and the GBT Recipient shall be the Access Control Broker and the HHT where such Remote Party Messages are also delivered to the HHT in line with the requirements of Section 10.5.3.2.

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

- in a Remote Party Command, the value in the Business Originator ID field; and
- in a Remote Party Response or a Remote Party Alert, the value in the Business Target ID field.

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

GBT Streaming Window shall be:

- 63 where the GBT Message Series carries a Remote Party Response;
- 6 where the GBT Message Series carries a Remote Party Command; or
- the number of GBT Messages the sender wishes to be resent in response to a GBT Message (Request Block Resend).

Maximum PDU Size shall be 1200 octets.

For clarity, recovery of lost blocks may take place before the end of the streaming window.

7.2.11.2 Remote Party Message size

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

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

7.2.11.3 GBT Message Structure

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

Message Routing Header || GBT Header

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

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

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

where:

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

7.2.11.4 GBT Message processing

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

1. GBT Block Data with GBT Initiator Block Number of 1 is the 1149 most significant octets of the Remote Party Message, or all of the octets if the size of the Remote Party Message is less than 1149 octets;
2. GBT Block Data with GBT Initiator Block Number of 2 is the next 1149 most significant octets of the Remote Party Message, or all of the octets if the size of the remaining octets in Remote Party Message is less than 1149 octets; and
3. remaining GBT Block Data are created by repeating Step 2, each time incrementing GBT Initiator Block Number by 1, until there are no remaining octets in the Remote Party Message.

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

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

When a GBT Message Series carries a Response and that GBT Message Series consists of more than 63 GBT Messages, the GBT Recipient shall, if it wishes to receive all parts of the GBT Message Series, send a GBT Message (Acknowledgement) when it has received each complete set of 63 sequential GBT Messages. Note, when the Device sending the Response receives such a GBT Message (Acknowledgement), it is able to begin sending the next sequence of GBT Messages in the GBT Message Series.

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

GBT Initiator Block Number Ack shall be the highest of:

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

GBT Recipient Block Number Ack shall:

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

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

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

For clarity, Devices shall discard malformed GBT messages without further processing in the sense of Section 6. For example, the following GBT Messages would be malformed:

- one with 'block-number' equal to zero (this does not comply with the GBCS); and
- one delivered to a recipient where 'block-number-ack' is greater than the greatest 'block-number' in any GBT Message the recipient has sent in that GBT Message Series.

Where the GBT Recipient is an ESME or an SAPC, the Device shall not send a GBT Message (Request Block Resend), in relation to a GBT Message Series, until and unless 30 seconds have elapsed since receipt of the most recent GBT Message in that GBT Message Series. If 30 minutes elapse without the Device receiving corresponding GBT Message(s) in response, the Device may discard the GBT Messages it has received in this GBT Message Series.

7.2.11.5 Message Routing Header

Message Routing Header				
No	**DLMS COSEM 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

Message Routing Header				
No	**DLMS COSEM Message Elements**	**Contents**	**Length (octets)**	**Note**
				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 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

Table 7.2.11.5: Message Routing Header

7.2.11.6 GBT Header

GBT Header				
No	**DLMS COSEM Message Elements**	**Contents**	**Length (octets)**	**Note**
	general-block-transfer	0xE0	1	DLMS COSEM 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

GBT Header				
No	**DLMS COSEM Message Elements**	**Contents**	**Length (octets)**	**Note**
				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 0 - 5)	See 'Note' column	6/8	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

Table 7.2.11.6: GBT Header

7.2.11.7 Illustrations – informative

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

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

<u>Example 1: A small Command with small Response – e.g. read MPAN</u>.

Without GBT, the Command is:

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

and the Response is:

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

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

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

and the Response becomes:

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

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

Without GBT, the Command is:

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

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

The Command is transmitted as the GBT Message Series:

Message Routing Header || GBT Header || block 1
Message Routing Header || GBT Header || block 2
Message Routing Header || GBT Header || block 3

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

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

The Response could have the following structure:

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

It will be smaller than the Maximum PDU Size and so can be sent as a single APDU without any use of GBT. In this case it would have the following structure:

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

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

The Command is:

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

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

The ESME Response is:

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

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

The ESME will send:

Message Routing Header || GBT Header || Block 1
…
Message Routing Header || GBT Header || Block 63

and wait for acknowledgement.

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

Message Routing Header || GBT Header

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

Message Routing Header || GBT Header || Block 64
…
Message Routing Header || GBT Header || Block 75

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

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

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

7.3 Device Requirements – DLMS COSEM

7.3.1 Introduction – informative

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

The Technical Specifications 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.

The Technical Specifications only require DLMS COSEM certification on the ESME and the SAPC.DLMS COSEM objects (or functionality equivalent to them) are required to deliver the ESME / SAPC functionality defined in the Use Cases in a consistent way but should not be accessible via the Device's HAN interface (i.e. it is internal functionality).

7.3.2 General Requirements

Constant values specified in Table 7.3.8a shall be fixed before operation and shall be immutable save via a firmware upgrade. This is to ensure consistent functioning and guard against potential attacks. Except where explicitly required by this Section 7.3, a Device shall not expose any part of any DLMS COSEM object, either for the writing of an attribute or for the invocation of a method that could, if used, constitute a Critical action.

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

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

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

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

- shall implement all of the DLMS COSEM objects, attributes and methods detailed in the 'SMETS Required Objects' tab of Table 20 which are:
  - marked 'mandatory' in the 'SAPC DLMS COSEM support' column; or
  - marked 'optional' in the 'SAPC support' column and are required for Use Cases that the SAPC does support,

and shall expose such specified attributes and methods over its network interface; and

- shall, where the SAPC supports the corresponding SMETS functionality, have the constant values set for the DLMS COSEM attributes specified as requiring constant values in Table 7.3.8a, and shall ensure that such values cannot be amended, save via activation of new firmware.

7.3.3 Application Associations

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

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

The Application Associations in Table 7.3.8c shall limit access to DLMS features by configuring the object_list attribute to reflect the access granted to the role in 'SMETS Required Objects' tab of the Mapping Table in Section 20. For SAPC, the object_list attribute settings shall limit access in line with the 'SMETS required objects' tab of Table 20 to only those those rows which are either:

- marked 'mandatory' in the 'SAPC DLMS COSEM support' column; or
- marked 'optional' in the 'SAPC DLMS COSEM support' column and are required for Use Cases that the SAPC does support.

Any other methods and attributes of any class shall be made inaccessible by listing them in the object_list attribute such that there is no access.

The Public AA shall only expose:

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

When a Message is received by the Device, the Message shall be validated against the AA based on the Business Originator ID and the Message Code within the Grouping Header.

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

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

7.3.4 Interface Classes and Objects

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

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

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

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

7.3.5 Values normally negotiated when an AA is established

7.3.5.1 Conformance Block Contents

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

7.3.5.2 Other Items.

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

7.3.5.3 Security Setup Objects

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

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

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

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

7.3.6 Scripts for operation of the ESME / SAPC

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

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

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

7.3.6.1 Auxiliary Controller related scripts

All italicised terms in this Section 7.3.6.1 shall have their DLMS COSEM meanings.

As required by DLMS COSEM, the *execute* method (*method 1)* of a *script table* object (*class_id = 9*) takes a *long-unsigned* value, so two octet value, as its parameter. To have an effect, the *value* of that *long-unsigned* parameter has to match one of the *script_identfiers* in that object's *scripts* attribute (*so attribute 2*). Similar requirements apply to the *script_selector* values in *schedule* objects (*class_id = 9*).

This Section 7.3.6.1 specifies the *script_identifier* values that SAPC and ESME shall support (and so the values such Devices shall accept in associated Use Cases) for *script table* objects with OBIS codes 0-0:10.0.103.255 and 0-0:10.0.107.255, and the functionality that such Devices shall implement when each such *script_identifier* in these two objects is executed.

SAPC and ESME shall support all *script_identifiers* in objects 0-0:10.0.103.255 and 0-0:10.0.107.255 where:

- the most significant nibble of the most significant octet has a value of either 0x1, indicating output to the controlled load, or 0x2, indicating input;
- the least significant nibble of the most significant octet has a value of either 0x1, 0x2, 0x3, 0x4 or 0x5, indicating 'n' in relation to Auxiliary Controller [n];
- for 0-0:10.0.103.255, the most significant bit of the least significant octet has a value of 0b0 or 0b1;
- for 0-0:10.0.107.255, the most significant bit of the least significant octet has a value of 0b0; and

- the seven least significant bits of the least significant octet have a value between 0b0000000 and 0b1100100 inclusive, so between 0 and 100.

SAPC and ESME shall undertake the following processing based on the value of *script_identifier*, when a *script* in these *script tables* is invoked:

- where the most significant octet has a value of 0x1n, the Device shall only undertake any further script invocation processing where there is currently an Auxiliary Controller associated with Auxiliary Controller [n] and shall only undertake processing in relation to that Auxiliary Controller;
- where the most significant octet has a value of 0x2n, the Device shall only undertake any further script invocation processing where there is currently an Auxiliary Proportional Controller associated with Auxiliary Controller [n] and shall only undertake processing in relation to that Auxiliary Proportional Controller;
- where the script invocation is being requested as part of an 'ECS47e Limit APC [n] Level' Command's execution, the Device shall only undertake any further script invocation processing where the currently associated Auxiliary Controller is an Auxiliary Proportional Controller;
- for 0-0:10.0.103.255, where the most significant bit of the least significant octet has a value of 0b1, the Device shall only take script invocation action where the *script* execution is being requested by the Auxiliary Controller Calendar (with its SMETS meaning) and so the *schedule* object with OBIS code 0-0:12.0.2.255;
- where the most significant bit of the least significant octet has a value of 0b0, the Device shall only take script invocation action where the *script* execution is being requested as part of an 'ECS47a Set Auxiliary Controller [n] State' or 'ECS47e Limit APC [n] Level' Command's execution (so where the value of the *data* parameter in the *execute* method / the controllerNumberAndCommandedState field meets this requirement);
- where the preceding conditions are all met, the Device shall undertake processing for the commanded state of the Auxiliary Controller [n] according to the SMETS requirements, using the following interpretation of the seven least significant bits of the least significant octet and, where relevant, the most significant nibble of the most significant octet:
  - for an Auxiliary Proportional Controller (with its SMETS meaning), the value of the seven least significant bits of the least significant octet shall be interpreted as a percentage between 0% (so 0b0000000) and 100% (so 0b1100100), which shall relate:
    - to the output state, if the most significant nibble of the most significant octet is 0x1; or
    - to the input state, if the most significant nibble of the most significant octet is 0x2;
  - for an ALCS, the value of 0b1100100 in the seven least significant bits of the least significant octet shall be interpreted as meaning closure of the switch (so allowing energy to flow) and any other value shall be interpreted as meaning opening of the switch (so not allowing energy to flow); and
  - for HCALCS, the value of 0b1100100 in the seven least significant bits of the least significant octet shall be interpreted as meaning closure of the switch (so allowing energy to flow) and so requiring the 'Duty Cycle' parameter in any resulting Load Control Event command to be 0x64; any other value shall be interpreted as meaning opening of the switch (so not allowing energy to flow) and so requiring

the 'Duty Cycle' parameter in any resulting Load Control Event command to be 0x00.

Where that processing leads to a change in the commanded state of Auxiliary Controller [n], the Device shall update the Auxiliary Controller [n] State's commanded state (so the value of either '*p*' or *'q'* in Section 7.2.9.2 and, where the Alerts are required by SMETS, the value of 'resultingLevel' in each of Use Cases 'ECS101 Limit APC [n] Level Command processed' and 'ECS102 Limit APC [n] Level ended or cancelled') to reflect the current commanded state:

- where Auxiliary Controller [n] is an ALCS or HCALCS and its current commanded state is closure of the switch, '*p*' and *'q'* shall have the value 100. If its current commanded state is opening of the switch, '*p*' and *'q'* shall have the value 0; and
- where Auxiliary Controller [n] is an Auxiliary Proportional Controller, '*p*' shall have a value between 0 and 100 inclusive, reflecting the percentage to which its commanded output state is currently set, '*q*' shall have a value between 0 and 100 inclusive, reflecting the percentage to which its commanded input state is currently set.

Correspondingly, as part of the step 4 checks in Section 6.2.4.1.1, Devices shall, for:

- *script_selector* values in Use Case 'ECS46d Set Auxiliary Controller Calendar' Commands; and,
- *data* values in the *execute method* of Use Case 'ECS47a Set Auxiliary Controller [n] State' / 'ECS47e Limit APC [n] Level' Commands,

apply the following validation to each of those values:

- the most significant nibble of the most significant octet has a value of 0x1 or 0x2;
- the least significant nibble of the most significant octet has a value of either 0x1, 0x2, 0x3, 0x4 or 0x5;
- the seven least significant bits of the least significant octet has a value between 0b0000000 and 0b1100100 inclusive, so between 0 and 100; and
- for Use Case 'ECS46d Set Auxiliary Controller Calendar', the most significant bit of the least significant octet is 0b1; and
- for Use Cases 'ECS47a Set Auxiliary Controller [n] State' and 'ECS47e Limit APC [n] Level', the most significant bit of the least significant octet is 0b0.

When creating an entry in the Auxiliary Controller Event Log (with its SMETS meaning) which relates to the execution of a script in objects 0-0:10.0.103.255 and 0-0:10.0.107.255, the Device shall record the *script_identifier* value in the entry_auxiliaryControllerLogEntry.switchNumberAndAction field of that log entry.

7.3.7 Pricing matrices, scripts and registers

Where an SAPC supports the related SMETS functionality, statements and requirements in this Section 7.3.7 shall apply to the SAPC as if it were an ESME.

7.3.7.1 Summary of approach – informative

As required by SMETS, an ESME has:

- a 1 * 48 matrix of primary element TOU (Time Of Use) consumption registers, and an associated 1 * 48 matrix of consumption based prices;
- a 4 block by 8 time band matrix of primary element 'TOU with Blocks' consumption registers, and an associated 4 * 8 matrix of consumption based prices;

- a tariff switching table, which specifies time based switching between primary consumption registers and so between different prices. (Switching between blocks is based on consumption passing thresholds and not on time);
- for Twin Element ESME only, a 1 * 4 matrix of secondary element TOU consumption registers, and an associated 1 * 4 matrix of consumption based prices; and
- for Twin Element ESME only, a secondary tariff switching table, which specifies time based switching between registers and so between different consumption based prices.

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

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

7.3.7.2 ESME requirements

An ESME shall:

- in calculating the cost of Consumption, for a Consumption Register specified in Table 7.3.7.2 apply the charge_per_unit in the charge_table_element identified by corresponding Index specified in Table 7.3.7.2 of the unit_charge_active attribute of the corresponding Charge Object specified in Table 7.3.7.2; and
- reject any instruction to set the unit_charge_passive attribute of a Charge Object in Table 7.3.7.2 that does not include all of the charge_table_elements specified in Table 7.3.7.2 for that Charge Object or contains charge_table_elements with values of index 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	0-0:19.20.0.255	0x01	
TOU(2)	0x0002	1-0:1.8.2.255	0-0:19.20.0.255	0x02	
TOU(3)	0x0003	1-0:1.8.3.255	0-0:19.20.0.255	0x03	
TOU(4)	0x0004	1-0:1.8.4.255	0-0:19.20.0.255	0x04	
TOU(5)	0x0005	1-0:1.8.5.255	0-0:19.20.0.255	0x05	
TOU(6)	0x0006	1-0:1.8.6.255	0-0:19.20.0.255	0x06	
TOU(7)	0x0007	1-0:1.8.7.255	0-0:19.20.0.255	0x07	
TOU(8)	0x0008	1-0:1.8.8.255	0-0:19.20.0.255	0x08	
TOU(9)	0x0009	1-0:1.8.9.255	0-0:19.20.0.255	0x09	
TOU(10)	0x000A	1-0:1.8.10.255	0-0:19.20.0.255	0x0A	
TOU(11)	0x000B	1-0:1.8.11.255	0-0:19.20.0.255	0x0B	
TOU(12)	0x000C	1-0:1.8.12.255	0-0:19.20.0.255	0x0C	
TOU(13)	0x000D	1-0:1.8.13.255	0-0:19.20.0.255	0x0D	
TOU(14)	0x000E	1-0:1.8.14.255	0-0:19.20.0.255	0x0E	
TOU(15)	0x000F	1-0:1.8.15.255	0-0:19.20.0.255	0x0F	
TOU(16)	0x0010	1-0:1.8.16.255	0-0:19.20.0.255	0x10	
TOU(17)	0x0011	1-0:1.8.17.255	0-0:19.20.0.255	0x11	

Description	Script Selector (long unsigned)	Consumption Register	Charge Object	Index (octet-string(1))	Notes
TOU(18)	0x0012	1-0:1.8.18.255	0-0:19.20.0.255	0x12	
TOU(19)	0x0013	1-0:1.8.19.255	0-0:19.20.0.255	0x13	
TOU(20)	0x0014	1-0:1.8.20.255	0-0:19.20.0.255	0x14	
TOU(21)	0x0015	1-0:1.8.21.255	0-0:19.20.0.255	0x15	
TOU(22)	0x0016	1-0:1.8.22.255	0-0:19.20.0.255	0x16	
TOU(23)	0x0017	1-0:1.8.23.255	0-0:19.20.0.255	0x17	
TOU(24)	0x0018	1-0:1.8.24.255	0-0:19.20.0.255	0x18	
TOU(25)	0x0019	1-0:1.8.25.255	0-0:19.20.0.255	0x19	
TOU(26)	0x001A	1-0:1.8.26.255	0-0:19.20.0.255	0x1A	
TOU(27)	0x001B	1-0:1.8.27.255	0-0:19.20.0.255	0x1B	
TOU(28)	0x001C	1-0:1.8.28.255	0-0:19.20.0.255	0x1C	
TOU(29)	0x001D	1-0:1.8.29.255	0-0:19.20.0.255	0x1D	
TOU(30)	0x001E	1-0:1.8.30.255	0-0:19.20.0.255	0x1E	
TOU(31)	0x001F	1-0:1.8.31.255	0-0:19.20.0.255	0x1F	
TOU(32)	0x0020	1-0:1.8.32.255	0-0:19.20.0.255	0x20	
TOU(33)	0x0021	1-0:1.8.33.255	0-0:19.20.0.255	0x21	
TOU(34)	0x0022	1-0:1.8.34.255	0-0:19.20.0.255	0x22	
TOU(35)	0x0023	1-0:1.8.35.255	0-0:19.20.0.255	0x23	
TOU(36)	0x0024	1-0:1.8.36.255	0-0:19.20.0.255	0x24	
TOU(37)	0x0025	1-0:1.8.37.255	0-0:19.20.0.255	0x25	
TOU(38)	0x0026	1-0:1.8.38.255	0-0:19.20.0.255	0x26	
TOU(39)	0x0027	1-0:1.8.39.255	0-0:19.20.0.255	0x27	
TOU(40)	0x0028	1-0:1.8.40.255	0-0:19.20.0.255	0x28	
TOU(41)	0x0029	1-0:1.8.41.255	0-0:19.20.0.255	0x29	
TOU(42)	0x002A	1-0:1.8.42.255	0-0:19.20.0.255	0x2A	
TOU(43)	0x002B	1-0:1.8.43.255	0-0:19.20.0.255	0x2B	
TOU(44)	0x002C	1-0:1.8.44.255	0-0:19.20.0.255	0x2C	
TOU(45)	0x002D	1-0:1.8.45.255	0-0:19.20.0.255	0x2D	
TOU(46)	0x002E	1-0:1.8.46.255	0-0:19.20.0.255	0x2E	
TOU(47)	0x002F	1-0:1.8.47.255	0-0:19.20.0.255	0x2F	
TOU(48)	0x0030	1-0:1.8.48.255	0-0:19.20.0.255	0x30	
Block(1)TOU(1)	0x0065	1-1:1.8.1.255	0-0:19.20.0.255	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	0-0:19.20.0.255	0xA2	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(1)TOU(3)	0x0067	1-1:1.8.3.255	0-0:19.20.0.255	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	0-0:19.20.0.255	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	0-0:19.20.0.255	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	0-0:19.20.0.255	0xA6	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(1)TOU(7)	0x006B	1-1:1.8.7.255	0-0:19.20.0.255	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	0-0:19.20.0.255	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	0-0:19.20.0.255	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	0-0:19.20.0.255	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	0-0:19.20.0.255	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	0-0:19.20.0.255	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	0-0:19.20.0.255	0xB5	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(2)TOU(6)	0x0072	1-2:1.8.6.255	0-0:19.20.0.255	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	0-0:19.20.0.255	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	0-0:19.20.0.255	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	0-0:19.20.0.255	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	0-0:19.20.0.255	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	0-0:19.20.0.255	0xC3	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
Block(3)TOU(4)	0x0078	1-3:1.8.4.255	0-0:19.20.0.255	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	0-0:19.20.0.255	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	0-0:19.20.0.255	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	0-0:19.20.0.255	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	0-0:19.20.0.255	0xC8	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(4)TOU(1)	0x007D	1-4:1.8.1.255	0-0:19.20.0.255	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	0-0:19.20.0.255	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	0-0:19.20.0.255	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	0-0:19.20.0.255	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	0-0:19.20.0.255	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	0-0:19.20.0.255	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	0-0:19.20.0.255	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	0-0:19.20.0.255	0xD8	Which block is activated by this Script Selector will depend on ESME consumption since last block reset
TOU(1) (Secondary Element)	0x00C9	1-20:1.8.1.255	0-0:19.20.5.255	0x01	Only present on Twin Element ESME
TOU(2) (Secondary Element)	0x00CA	1-20:1.8.2.255	0-0:19.20.5.255	0x02	Only present on Twin Element ESME
TOU(3) (Secondary Element)	0x00CB	1-20:1.8.3.255	0-0:19.20.5.255	0x03	Only present on Twin Element ESME
TOU(4) (Secondary Element)	0x00CC	1-20:1.8.4.255	0-0:19.20.5.255	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

Table 7.3.8a: Objects tab in embedded file

Table 7.3.8b: Scripts tab in embedded file

Table 7.3.8c: Application Associations tab in embedded file

Table 7.3.8d: Association LN Object Content tab in embedded file

Table 7.3.8e: Security Setup Object Content tab in embedded file

Table 7.3.8f: SAP Assignment Object content tab in embedded file

Table 7.3.8g: Conformance Content tab in embedded file

Table 7.3.8h: End to End Communications tab in embedded file

GBCS v4.0 Table 7.3.8.xlsx

7.3.9 ESME / SAPC accounts, credits and charges – informative

Where an SAPC supports the related SMETS functionality, the statements in this Section 7.3.9 apply to the SAPC as if it were an ESME.

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

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

Both relate to import energy.

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

The 'Set Payment Mode to Credit' and 'Set Payment Mode to Prepayment' Use Cases are used to trigger such activation of the 'passive' object. The SMETS attributes Suspend Debt Emergency and Suspend Debt Disabled are implemented as part of these objects and so are set in these Use Cases. If an ESME is in Prepayment Mode and the value of either Suspend Debt attribute is to be changed, this can be achieved by sending a Set Payment Mode to Prepayment Command containing the new values.

7.3.10 ESME / SAPC requirements for using Special Days objects

When applying Blue Book special days related requirements to a Calendar / Scheduler object listed in Table 7.3.10, an ESME and an SAPC, where it supports the related SMETS functionality, shall use the corresponding Specials Days Object in Table 7.3.10.

	Calendar / Scheduler object		*Special Days Object to be used*
SMETS Reference	***Class ID***	***OBIS***	***OBIS***
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
AuxiliaryControllerCalendar(SpecialDays)	10	0-0:12.0.2.255	0-0:11.0.3.255

Table 7.3.10: Special Days Object

7.4 Device requirements – ZSE

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

Only Devices capable of operating at Sub-GHz shall be required to support the requirements in rows of Table 7.4 where the cell in the column labelled 'Sub GHz capable Devices only?' contains 'Yes'.

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

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

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

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

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

An SAPC shall implement a *ZSE Energy Services Interface* and shall implement all the *clusters, commands, attribute sets and attributes* in Table 7.4 where column A is 'SAPC: Energy Services Interface (Electricity ESI Endpoint)'.

Where it supports the corresponding SMETS functionality, an SAPC shall implement the *clusters, commands, attribute sets and attributes* in Table 7.4 where column A is 'SAPC optional: Energy Services Interface (Electricity ESI Endpoint)'.

Additionally, an SAPC may support other *clusters, commands, attribute sets and attributes* in Table 7.4 where column A is 'ESME: Energy Services Interface (Electricity ESI Endpoint)'.

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

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

1. the first which shall implement all the *clusters, commands, attribute sets and attributes* in Table 7.4 where column A is 'ESME: Energy Services Interface (Twin ESME aggregate ESI Endpoint)';
2. the second which, in relation to the primary 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)'; and

3. 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)'.

A PPMID shall implement a *ZSE In-Home Display*, shall implement all the *clusters, commands, attribute sets and attributes* in Table 7.4 where column A is 'PPMID: In-Home Display', and shall support the other clusters, attributes and commands necessary to meet the SMETS requirements.

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

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

An IHD shall implement all the clusters, commands, attribute sets and attributes in Table 7.4 where column A is 'IHD: In-Home Display' and shall support the other clusters, attributes and commands necessary to meet the SMETS requirements.

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

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

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

Except where explicitly required by this Section 7.4 or by Section 19.3, a Device shall not execute any ZSE command, be that in a GBZ Command Payload or provided as a native ZSE command, that could, if executed, constitute a Critical action. For clarity, a Device shall not execute a ZSE *Publish Change of Supplier* command if bits 11-12 of the *Provider Change Control* parameter (*Meter Contactor State*) of that command has any value other than 0b11 (*Supply UNCHANGED*).

In relation to the *calendar cluster's* use for exchanging information about the Auxiliary Controller Calendar (with its SMETS meaning) between Devices:

- the *Calendar Type* value of 0x04 (named *Auxillary Load Switch Calendar*) shall be used, to differentiate from the values of 0x00 (named *Delivered Calendar*) and 0x03 (named *Friendly Credit Calendar*) for the Tariff Switching Table and Non-Disablement Calendar, respectively (with their SMETS meanings); and
- the *Auxiliary Load Switch State* parameters in *PublishDayProfile* commands shall be set so that:
  - *bit0* to *bit4* have values corresponding to Auxiliary Controller [1] to Auxiliary Controller [5] respectively, with each such bit being 0b1 where the commanded output state is 100 for the switching instruction in question (so the value of 'p' with its Section 7.3.6.1 meaning), or 0b0 otherwise; and
  - *bit5* to *bit7* have the value 0b0.

  For clarity, this means that settings related to commanded input states are not shared, and commanded output settings of 99 or less on an Auxiliary Proportional Controller (with its SMETS meaning) are all represented as 0b0 to other HAN Devices.

SECMP0007-DP164-Markup___Table_7-4_ GBCS v4.1 Table 7.4.xlsx

Field Code Changed

Table 7.4: 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 or Consumption Register. Each of these is a single integer value;
- the Supplier reads the values held in the Tariff Block Counter Matrix, Tariff TOU Register Matrix or Tariff TOU Block Register Matrix;
- 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, the Daily Read Log (excluding the export related parts), 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);
  - the Network Operator reads the Network Data Log;
- 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.

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

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

8.3 Key Derivation Inputs

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

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

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

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

8.4 AAD, Plaintext and Ciphertext

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

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

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

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

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

Ciphered Information shall be the concatenation:

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

8.4.1 Meaning of SC – informative

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

- Bits 3..0 are security suite which is 0b0001 since Security Suite 1 is required;
- Bit 4 is set to 0b1 since Authentication of the data is required;

- Bit 5 is set to 0b1 since Encryption of the data is required;
- Bit 6 is set to 0b0 since Messages containing the encrypted data are unicast; and
- Bit 7 is set to 0b0 as per the Green Book.

8.5 Access to sensitive data – COSEM attribute access

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

For clarity and in line with the Blue Book, the first octet of the plaintext input, constructed by the Device when using the Data Protection class, shall be 0x02 (the tag for structure according to Section 9.5 of the Green Book) and the next octet shall be the number of captured DLMS COSEM attributes in the Use Case.

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

8.5.1 Values of the Data Protection class attributes

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

- protection_object_list (attribute 3) shall be an array containing object_definition(s). Within an object_definition: class_id, logical_name, attribute_index, data_index, restriction_type and restriction_value shall take values as per Table 7.3.8a;
- the value of protection_parameters_get (attribute 4) shall be a single entry array containing one protection_parameters_element, which shall have the values specified in Table 8.5.1;
- the value of protection_parameters_set (attribute 5) shall be an array containing zero entries; and
- the value of required_protection (attribute 6) shall be 0b01100000 (0x60) where the object exposes the get_protected_attributes(data) method, since Authenticated Encryption is required on the output of the method.

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

Table 8.5.1: Values of protection_parameters_element

8.5.2 Parameters of the get_protected_attributes method

The get_protected_attributes_request parameter of the get_protected_attributes method shall:

- be populated in the Command to the Device according to Table 8.5.2a; and
- be verified by the Device receiving the Command according to Table 8.5.2a.

The protection_parameters part of the get_protected_attributes_response returned by the get_protected_attributes method shall be populated by the Device according to Table 8.5.2b.

The value of protected_attributes part of the protected_attributes_response_data returned by the get_protected_attributes method shall be populated by the Device with Ciphered Information, calculated as per the requirements of Section 8.2. The tag for protected_attributes shall be 'octet-string' (0x09) and the length shall be the length of 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	See 'Note' column	Must have this value	Number of entries in the array
object_definition			Entry in the object_list array. May be more than 1 occurrence
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

Field	Value	Device Validation	Note
value	See 'Note' column	Must be the same as the logical_name in attribute 3 of the Data Protection object being accessed	The logical_name of the object which is the source of the Encrypted data
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	If this is to access a Log as defined in Section 8.2

Field	Value	Device Validation	Note
		Log as defined in Section 8.2	
restriction_type			
tag	0x16	Must have this value	Meaning 'enum'
value	0x01	Must have this value	Meaning 'restriction by date'
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'

Field	Value	Device Validation	Note
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 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'

Field	Value	Device Validation	Note
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'
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 8.5.2a: values of get_protected_attributes_request

Field	Value	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'

Field	Value	Validation	Note
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 note		The Entity Identifier of the Device
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		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	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'
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

8.5.3 Billing Data Log Alert – DLMS COSEM

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

8.6 Access to sensitive data – ZSE attribute access

Ciphered Information shall be used to populate each 'GBZ Use Case Specific Component 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 SAPC, 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 SAPC, 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 Device. 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 and SAPC as specified in Section 9.1.4, and GSME as specified in Section 9.1.7.

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.

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

Requirements in this Section 9.1.3 shall apply to an SAPC as if it were an ESME.

All italicised terms in this Section 9.1.3 shall have the meanings defined in the *Time Cluster* specification within the ZigBee Cluster Library Specification.

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;
- whenever its Time attribute has the value 0xFFFFFFFF, store the value 0xFFFFFFFF in any unsigned 32 bit integer format time stamps it records and store the value 0xFFFFFFFFFFFFFFFFFF8000FF in any DLMS COSEM octet-string(12) format time stamps it records; 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 no valid Time has been provided after three retries, the Device shall set its Time Status to 'Unreliable';
- if the time provided by the Communications Hub differs from the Device's time by more than 10 seconds, then the Device shall:
  - set its Time Status to 'Unreliable'; and
  - if this results in a change to Time Status, the Device shall construct and issue an Alert with Alert Code 0x8F0C, meaning that its time would have been shifted by more than 10 seconds. If the Device is a GSME, the Alert Payload shall be a GBZ Alert Payload. If the Device is an ESME, the Alert Payload shall be a DLMS COSEM Alert Payload.
- if the time provided by the Communications Hub differs from the Device's time by 10 seconds or less, then the Device shall adjust its time to the Communications Hub's time.

9.1.4 ECS70 Set Clock on ESME

Requirements in this Section 9.1.4 and in Use Case 'ECS70 Set Clock on ESME' shall apply to an SAPC as if it were an ESME.

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
Service 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

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

9.1.4.1 Pre-conditions

None.

9.1.4.2 Detailed Steps

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 0xFFFFFFFFFFFFFFFFFF8000FF 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

Table 9.1.4.2a: Construction of Command Payload

In this Section 9.1.4.2, the object with OBIS code 0-0:1.0.0.255 shall be referred to as the *Clock* and italicised terms shall have their Blue / Green Book meaning.

On receipt of the Command, the ESME shall undertake processing in the following sequence:

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

ii. if the current Time Status is not set to 'Reliable', the ESME shall not change status.

b) the *preset_adjusting_time* method of the Clock shall be executed. Note this is to set parameters for the *adjust_to_present_time* method;

c) the adjust_to_present_time method of the Clock shall be executed as follows:

iii. if the Communications Hub Time returned lies between *validity_interval_start* and *validity_interval_end*, then:

a. ESME *time* shall be updated to match Communications Hub Time;

b. the ESME shall unset bit 0 of attribute 4 (so setting *status* of the *Clock* not to be an '*invalid value*'); and

c. the ESME shall unset bit 1 of attribute 4 (so setting *status* of the *Clock* not to be a '*doubtful value*').

iv. if the Communications Hub Time returned lies before *validity_interval_start* (Supplier Current Time) or after *validity_interval_end* (Supplier Current Time + Tolerance)) and (bit 0 of attribute 4 of the *Clock* is unset), then:

d. time shall remain unchanged, since time is outside the *validity_interval*; and

e. the ESME shall set bit 1 of attribute 4 and unset bit 0 of attribute 4 (so setting status of *Clock* to be a '*doubtful value*' (Unreliable)).

d) the *get request* on the *time* and *status* attributes of the *Clock* shall be executed.

3. the ESME shall undertake the 'Response Construction' and 'Response Cryptographic' processing required for a Response of type SME.C.C.

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

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

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

Values in attribute 4 of the *Clock* 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

Table 9.1.4.2b: Meaning of unsigned integer

9.1.5 Time related object on ESME

Requirements in this Section 9.1.5 shall apply to an SAPC as if it were an ESME.

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

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

- shall be a double-long-unsigned value;
- shall have a value set by the ESME to the number of seconds between 0 hours 0 minutes 0 seconds on 1st January 2000 UTC and the value of UTC time specified by attribute 2 of the *Clock* object with OBIS code 0-0:1.0.0.255;

- shall be the value recorded by the ESME in attribute 2 of any *Profile generic* object *entry* as the date-time stamp, at the time the *entry* is added;
- shall be the format recorded by the ESME in attribute 2 of any *Profile generic* object *entry* in other date-time fields.

Correspondingly:

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

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

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

9.1.6 Start of Time and End of Time values

Where a date-time is specified as a 32 bit long unsigned integer:

- the Start of Time shall mean the value 0x00000000; and
- the End of Time shall mean the value 0xFFFFFFFF.

When a date-time is specified in the DLMS COSEM octet-string(12) format:

- the Start of Time shall mean the value 0x00000000000000000008000FF; and
- the End of Time shall mean the value 0xFFFFFFFFFFFFFFFFFFF8000FF'.

9.1.7 GCS28 Set Clock on GSME

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
Service 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

Cross Reference	Value
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

Table 9.1.7: Use Case Cross References for GCS28 Set Clock on GSME

9.1.7.1 Pre-conditions

None.

9.1.7.2 Construction of Command

Set Clock Command Payloads shall be constructed according to the requirements of Section 9.1.7.4 and populated as specified in Table 9.1.7.2.

MAC Header, Grouping Header, KRP Signature and ACB-SMD MAC shall be populated as 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

Table 9.1.7.2: `@SetTime.CommandPayload` population

9.1.7.3 Device processing of Command and Response handling

The GSME receiving a Set Clock Command shall undertake processing steps in the sequence defined in this Section 9.1.7.3.

The GSME shall:

1. undertake Command Authenticity and Integrity Verification as required for a Command of the SME.C.C Message Category;
2. request the now current Communications Hub Time. If the Communications Hub cannot supply a valid time, it shall provide 0xFFFFFFFF. If this is sent, or no response is received, the GSME shall:
   a) if its current Time Status is set to 'Reliable', set Time Status to 'Unreliable', set `deviceTimeStatus` to `unreliable`, populate `deviceTime` with its current Time and process from step 5; or
   b) if its current Time Status is not set to 'Reliable', set `deviceTimeStatus` to be Time Status, populate `deviceTime` with its current Time and process from step 5.
3. if ((the Communications Hub Time < `validityIntervalStart`) or (Communications Hub Time > `validityIntervalEnd`)), set Time Status to 'Unreliable', and set

deviceTimeStatus to unreliable, leave its Time unchanged, populate deviceTime with its current Time and process from step 5;

4. set its time to the Communications Hub Time, populate deviceTime with the corresponding value, and set deviceTimeStatus to reliable;
5. populate the Response Payload according to the requirements of Section 9.1.7.4 using the deviceTimeStatus and deviceTime values produced by the processing in this Section 9.1.7.3;
6. construct Grouping Header and apply the Response Cryptographic Protection required for a Response of the SME.C.C Message Category; and
7. send the Response.

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

9.1.7.4 Set Clock Command and Response Payloads – structure definition

Each instance of @SetTime.CommandPayload and of @SetTime.ResponsePayload shall be an octet string containing the DER encoding of the populated structure defined in this Section 9.1.7.4 which specifies the structure in ASN.1 notation.

```
SetTime DEFINITIONS ::= BEGIN


CommandPayload ::=                                  SEQUENCE
{
     -- specify the period within which the Communications Hub's time must lie
     -- if this Command is successfully to set time
     validityIntervalStart                          GeneralizedTime,
     validityIntervalEnd                            GeneralizedTime

}

ResponsePayload ::=                                 SEQUENCE

{
     -- Specify the Device's now current time
     deviceTime                                     GeneralizedTime,

     -- Specify the Device's now current Time Status
     deviceTimeStatus                               DeviceTimeStatus
}

DeviceTimeStatus ::= INTEGER
{
     reliable                                       (0),
     invalid                                        (1),
     unreliable                                     (2)
}

END
```

9.2 Future Dated Remote Party Messages

9.2.1 Future Dated Commands for the Reading of Data Items – informative

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

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

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

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

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

9.2.2 Future Dated Commands for the Writing of Attributes

9.2.2.1 Introduction – informative

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

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

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

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

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

1. a Command would be sent to the Device instructing that the data items in question should be stored in the 'next' data items and the corresponding activation date / time should be set (so overwriting previous 'next' values and any previous activation date / time);
2. only if the Command is Authenticated, would the Device attempt to execute the instructions in Command. Execution for such Commands means writing to 'next' values and setting activation date-times. If the activation date-times are in the past, the Device will also attempt to make the 'next' values 'current';
3. the Device would then create a Response. This Response would either confirm that the 'next' values and the activation date / time values have been set (and so any previous future dated command with this Message Code has been over written), or would provide failure reasons. The 'current' values would be unaffected assuming the activation date / times are in the future; and
4. when an activation date / time is reached (according to the clock on the Device), the Device would attempt to make the corresponding 'next' values 'current'. The Device would then send an Alert detailing success or failure.

Like all other Commands, future dated Commands cannot be modified once accepted by the Device. However, the time activated processing can be stopped from happening by sending

a new Command of the same Message Type. This is because the new Command over writes the values from the old Command.

For example:

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

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

When there is a change of Supplier on a Device which is (1) after a future dated change is stored but (2) before it is activated, the processing at Section 13.3.5.10 will be undertaken at the point of update of Security Credentials. This ensures future dated commands from the old Supplier will not be actioned by the Device.

9.2.2.2 Date-times in future datable commands

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

9.2.2.3 Effect on prior Commands of the same Message Code

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

9.2.2.4 Using a future dated Command to write Attributes immediately

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

- for DLMS COSEM Commands Payloads, activation date-time(s) have the value 0x0000000000000000008000FF;
- for ZSE Command Payloads, the activation date-time(s) have the value 0x00000000; and
- for ASN.1 Command Payloads, the activation date-time is not present.

9.2.2.5 Cancellation of future dated Commands for the writing of Attributes

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

- for DLMS COSEM Commands Payloads, activation date-time(s) have the value 0xFFFFFFFFFFFFFFFFFF8000FF;
- for ZSE Command Payloads, the activation date-time(s) have the value 0xFFFFFFFF for any ZSE command other than PublishCalendar, PublishSpecialDays, PublishBlockThresholds, PublishPriceMatrix;

- for ZSE Command Payloads, the activation date-time(s) have the value 0xFFFFFFFE for the ZSE command PublishCalendar, PublishSpecialDays, PublishBlockThresholds, PublishPriceMatrix; and
- for ASN.1 commands, the activation date-time has a value of 99991231235959Z.

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

9.2.2.6 Reactions to Future Dated Commands

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

- where activation date-times are in the past or the instructions detail immediate execution as per Section 9.2.2.4, the Command shall be executed immediately:
  - for a CS02b Update Security Credentials Command, the processing shall be as required by Section 13.3.4.3;
  - for a CS02g Update Load Controller Security Credentials Command, the processing shall be as required by Section 13.10.3.4;
  - for a CS06 Activate Firmware, the processing shall be as required by Section 11.5.2.2; and
  - for all other Commands, the Response shall detail the outcome of the Command's execution, and no Alert shall be generated;
- where activation date-times are in the future, that Response shall detail the success or otherwise of storing the details in the Command; and
- if the activation date-times are in the future and the Command's details were successfully stored, the Device shall, at the time the future activation date-time is reached, process each of the instructions, which contain an activation date-time, as specified in the Command in the sequence specified in that Command and then generate an Alert with an Alert Payload, for each instruction, of the same type as the Payload type of the Command and an Alert Code of 0x8F66 for successful execution and 0x8F67 for failed execution. Thus:
  - an ASN.1 Command Payload shall lead to an ASN.1 Alert Payload, which shall be as defined in Sections 11 and 13;
  - a DLMS COSEM Command Payload shall lead to DLMS COSEM Alert Payload(s), which shall be as defined in Table 7.2.9c, where the Use Case specific additional content contains the concatenation 0x09 || 0x13 || Message Code || Originator Counter || cosem-attribute-descriptor from the corresponding part of the Command Payload. Note that 0x09 is the DLMS COSEM tag for octet-string and 0x13 is the length of the concatenation Message Code || Originator Counter || cosem-attribute-descriptor; or
  - a GBZ Command Payload shall lead to GBZ Alert Payload(s), shall be as defined in Table 7.2.10c, where the Use Case specific additional content contains the concatenation 0x0E || Message Code || Originator Counter || Extended Header Cluster ID || Frame control || Command identifier from the corresponding part of the Command payload.

9.2.2.7 ESME requirements for activation of future datable Commands

Requirements in this Section 9.2.2.7 shall apply to an SAPC as if it were an ESME, where either:

- the requirements relate to the Auxiliary Control Calendar, with its SMETS meaning; or
- the requirements relate to SMETS functionality which the SAPC supports.

When either (1) a Command successfully sets the 'passive' *Account* object's *account_activation_time (attribute ID 13, OBIS code 0-1:19.0.0.255*) to 0x00000000000000000008000FF or (2) the ESME's clock reaches the value in that attribute or (3) the *activate_account* method is invoked, the ESME shall not activate 'passive' *Account* object, as detailed in the Blue Book, but shall set the attributes in the 'active' *Account* object (*OBIS code* 0-0:19.0.0.255) as follows:

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

When either (1) a Command successfully sets an Activation Date-Time Attribute in Table 9.2.2.7 to 0x00000000000000000008000FF or (2) the ESME's clock reaches the value in an Activation Date-Time Attribute in Table 9.2.2.7, the ESME shall set the corresponding Active Attribute in Table 9.2.2.7 to the value of the corresponding Passive Attribute in Table 9.2.2.7.

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
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
TariffSwitchingTable(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	0-1:12.0.1.255	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
AuxiliaryControllerCalendar	9000	0-0:94.44.128.26	6	10	0-1:12.0.2.255	2	10	0-0:12.0.2.255	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
AuxiliaryControllerCalendar(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

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

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

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

The *ProtocolID* of all Remote Party Messages shall be 6. Devices shall set the value of the *ProtocolID* field in any *RequestTunnel* command to 6.

Devices shall set the value of the *FlowControlSupport* field in any *RequestTunnel* command to ‘*False*’.

All Devices except Type 2 Devices and GPFs shall support the *Tunneling Cluster* and, within that Cluster, the use of the protocol with a *ProtocolID* of 6.

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

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

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

A GPF shall support mirroring functionality. The *Basic Cluster Physical Environment* attribute shall, contrary to ZSE, be supported and shall have the value 0x01.

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

Where a Device is required to support the Tunneling Cluster as a server, the server Device shall:

- when sending a TransferData command, use the last TunnelID which it holds and reset the inactivity counter accordingly. If it receives a TransferDataError command in response with a TransferDataStatus of 0x00 (No such tunnel), it shall dis-regard the last TunnelID and close the associated tunnel;
- persist TunnelID values, and the associated client Device’s short address and endpoint, including when power is lost; and
- when receiving a RequestTunnel command from a client Device which has an active TunnelID registered with the server Device, reset the corresponding inactivity timer for that active TunnelID and send a RequestTunnelResponse containing that activeTunnelID.

Where a Device is required to support the Tunneling Cluster as a client, the client Device shall:

- whenever it sends a TransferDataError command with a TransferDataStatus of 0x00 (‘No such tunnel) to a server Device, then send a RequestTunnel command to that server Device; and
- whenever it receives a TransferDataError command with a currently valid TunnelID and a TransferDataStatus of 0x00 (‘No such tunnel) from a server Device, close any associated tunnel and send a RequestTunnel to that server Device.

10.2.2.1 Tunneling Requirements

As soon as possible after a Device, which is required to support the *Tunneling* Cluster as a client, has successfully established a shared secret key using *CBKE* with a Device, which is required to support the *Tunneling* Cluster as a server, the *tunneling* client Device shall send a *RequestTunnel* command to the *tunneling* server Device to request a *tunnel* association with the Device.

Where such a *tunneling* server Device remains in the *tunneling* client's Device Log, the *tunneling* client Device shall send a *RequestTunnel* command to the Device whenever 0xFFFF seconds have elapsed since receipt of the most recent *RequestTunnelResponse* command from that Device.

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

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

10.2.2.2 GSME

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

- send a request to the *ZigBee Gas ESI Endpoint* requesting the creation of mirrored *Basic, Metering* and *Prepayment Clusters* using the *RequestMirror* command;
- configure, using the *ConfigureMirror* command, the *ZigBee Gas Mirror Endpoint* to use the two way mirroring notification scheme '*Predefined Notification Scheme B'* ; and
- send a *RequestTunnel* command to the CHF to request a tunnel association with the CHF.

In line with ZSE, when a GPF sends a *RequestMirrorResponse* command in response to a *RequestMirror* command, the *RequestMirrorResponse* command shall contain the *EndPointID* to be used by the GSME regardless of whether the *RequestMirror* created the mirror.

A GPF shall only send a *RequestMirrorResponse* containing the *EndPointID* to the Device which caused the GPF to create the mirror.

Where a GPF receives a *ConfigureMirror* command to use the two way mirroring notification scheme '*Predefined Notification Scheme B*' which has the *Disable Default Response Sub-field* in its *Frame Control Field* set to zero, the GPF shall respond with a *Default Response* indicating *SUCCESS* if it has a mirror configured to use '*Predefined Notification Scheme B*', regardless of whether that was configured by the *ConfigureMirror* command.

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

Where a GSME reports a value for the *ManufacturerName* attribute or the *ModelIdentifier* attribute, the GPF shall accept that value. For clarity, there are no requirements for the GPF to subsequently process or make available any such value.

For clarity, the GSME:

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

Bit Number	Waiting Command
6	*Get Prepay Snapshot*
7	*Get Top Up Log*
9	*Get Debt Repayment Log*

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

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

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

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

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

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

- the GSME has created an Alert or Response that is to be sent; or
- the GSME has ascertained, via the *Tunnel Message Pending* flag, that there is a Command for it buffered on the Communications Hub.

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

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

Immediately following the successful establishment of the tunnel between the CHF and the SAPC / ESME / GSME, the SAPC / ESME / GSME shall send an Alert with Alert Code 0x8F69.

10.2.2.2.1 MirrorReportAttributeResponse command support

GPF shall:

- support *ConfigureMirror* commands where the '*Mirror Notification Reporting*' field is set to 0x01, and treat such a value as a request from the GSME for the GPF to push *FunctionalNotificationFlags* values using the *MirrorReportAttributeResponse* command;
- treat ZSE references to '*when the MirrorReporting attribute is set*' as references to '*when the Mirror Notification Reporting field in the most recently accepted ConfigureMirror command was set to 0x01*'; and
- send a *MirrorReportAttributeResponse* command to the GSME, containing the *FunctionalNotificationFlags* and *NotificationFlags#N* values required for '*Predefined Notification Scheme B*', whenever it receives a *ReportAttributes* command from the GSME for any '*Attribute Reporting Status*' attribute with a value of '*Attribute Reporting Complete*'. Note that the GSME may support one such '*Attribute Reporting Status*' attribute in each ZSE cluster.

Therefore, and for clarity, where a GSME wishes to receive *MirrorReportAttributeResponse* commands from the GPF, the GSME should:

- notify the GPF of that wish by setting the '*Mirror Notification Reporting*' field to 0x01 in *ConfigureMirror* commands it sends;
- support the '*Attribute Reporting Status*' attribute on each of the mirrored clusters, so *Basic*, *Metering* and *Prepayment*; and
- send a *ReportAttributes* command from the GSME for an *Attribute Reporting Status* attribute with a value of '*Attribute Reporting Complete*', each time it wishes to trigger the GPF to send a *MirrorReportAttributeResponse* command.

10.2.2.3 CHF requirements for ESME Transfer Data commands

Requirements in this Section 10.2.2.3 shall apply to an SAPC as if it were an ESME.

When a CHF is sending a Remote Party Message to an ESME in a *TransferData* command, the CHF shall:

- set the *Disable Default Response* sub-field of the *Frame Control* field in the *ZCL Header* of the *TransferData* command to 0b0;
- if it is the first time this Remote Party Message has been sent by the CHF, set a counter (the 'Message Transmission Counter') for this Remote Party Message to 1. Otherwise increase the Message Transmission Counter by 1; and
- not send any subsequent Remote Party Message to the ESME until either:
  - it has received a *Default Response* with a *Status Code* of *SUCCESS* in relation to a *TransferData* command containing the current Remote Party Message; or
  - it has sent an Alert with Code 0x8F84 ('Failure to Deliver Remote Party Message to ESME') in relation to the current Remote Party Message.

If the CHF receives a *Default Response* with a *Status Code* of *SUCCESS* to the *TransferData* command for the current Remote Party Message, the CHF may now send subsequent Remote Party Messages to the ESME.

If the CHF receives no *Default Response* after 10 seconds, or receives a *Default Response* with a *Status Code* other than *SUCCESS*, to the *TransferData* command for the current Remote Party Message, the CHF shall start the Flow Control Replay Timer from 0 seconds.

When the Flow Control Replay Timer reaches 30 seconds:

- if the Message Transmission Counter for the current Remote Party Message is less than 3, the CHF shall retry delivery of the current Remote Party Message, by sending it in a new *TransferData* command to the ESME; or
- if the Message Transmission Counter for the current Remote Party Message is 3, the CHF shall:
  - discard the current Remote Party Message; and
  - log the event in its Event Log and send an Alert with Alert Code 0x8F84 ('Failure to Deliver Remote Party Message to ESME') and Message Code 0x00D5, in relation to the current Remote Party Message.

10.2.2.4 Use Case for Alert: 'Failure to Deliver Remote Party Message to ESME'

Requirements in this Section 10.2.2.4, in all sub-Sections within it and in the Use Case for Alert: 'Failure to Deliver Remote Party Message to ESME' shall apply to an SAPC as if it were an ESME.

10.2.2.4.1 Description

This Section 10.2.2.4 covers the creation and processing of Alerts resulting from the requirements of Section 10.2.2.3.

10.2.2.4.2 Use Case Cross References

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Alert
Message Type Category	SME.A.C
Capable of future dated invocation?	N/A
Protection Against Replay Required?	N/A
Service Reference	N/A
Valid Initiating Device(s)	CHF
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	DLMS COSEM

Table 10.2.2.4.2: Use Case Cross References for Alert: 'Failure to Deliver Remote Party Message to ESME'

10.2.2.4.3 Construction of Alerts

Grouping Header and SMD Signature shall be populated as required for an Alert of the SME.A.C Message Category, with the Message Code being 0x00D5 and Alert Code being 0x8F84.

The Business Originator ID field in the Grouping Header shall contain the CHF's Entity Identifier.

The Business Target ID field in the Grouping Header shall contain the Entity Identifier in the CHF's `{accessControlBroker, digitalSignature, management}` Trust Anchor Cell.

The Use Case Specific Additional Content, with its Section 7.2.9 meaning, shall be an octet-string containing the value of the Message Identifier from the Remote Party Message that has not been delivered to the ESME, and so is the subject of this Alert.

Thus, the Use Case Specific Additional Content shall be the following concatenation, with values taken from the Remote Party Message that was not delivered:

0x09 || 0x19 || value of Business Originator ID || value of Business Target ID || value of CRA Flag || value of Originator Counter

10.2.2.4.4 Processing of Alerts

The SMD Signature may be verified by the Access Control Broker as per Section 6.7.4.

10.2.3 GSME Tunnel Management – informative

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

In common with the transport of all Remote Party or SME.C.PPMID-GSME Messages, the mechanism used is the *TransferData* command, but *TransferData* commands sent between a GSME and CHF need to distinguish between when:

- the GSME is sending a Message, so an Alert or a Response or a GBT Message containing part of an Alert / Response;
- the GSME is asking the CHF to send it a Command, or a GBT Message containing part of a Command; and
- the CHF is sending the GSME a Command, or a GBT Message containing part of a Command.

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

- where a Message is being sent, set the *Data* parameter payload in a *TransferData* command to the concatenation:

  Tunnel Manager Header || Message

- where a Message is not being sent (so when the GSME is requesting that a Message is sent), set the *Data* parameter payload in a *TransferData* command to the value of Tunnel Manager Header.

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

The ZSE specification has a flag called *Tunnel Message Pending* in the *Functional Flag Notification* definition. This flag is used to notify a GSME that the CHF has a Message waiting to be transferred to the GSME. The flag is set on the first pending Command and is reset when all Messages have been transferred to the GSME. The flag is available through the *ReadAttribute* or *MirrorAttributeResponse* command. The requirements for setting this flag are specified in Section 10.3.4.

The Tunnel Manager Header identifies three different kinds of *TransferData* command usage:

- GET (the value 0x01): this is used by the GSME to retrieve waiting Message from the CHF;
- GET-RESPONSE (the concatenation 0x02 || (Number of Messages remaining for retrieval after this Message) || (Message addressed to the GSME)): this is used by the CHF to send a Message to the GSME. It also indicates how many Messages have yet to be retrieved; and
- PUT(the value 0x03): this is used by the GSME to send a Message via the CHF.

Where a Command is waiting on the CHF for the GSME to retrieve it, the following sequence shall apply:

1. the *Tunnel Message Pending* flag is set on the Communications Hub as detailed in Section 10.3.4;
2. the GSME turns on its HAN Interface and obtains the value of the *Tunnel Message Pending* flag; and
3. If the *Tunnel Message Pending* flag is set:
    a) the GSME sends a *TransferData* command to the CHF with the GET structure in the Tunnel Manager Header. The Tunnel Manager Header is the only content in the *Data* field of this *TransferData* command;
    b) the CHF sends a *TransferData* command to the GSME with the GET-RESPONSE structure in the Tunnel Manager Header and a Message in the remaining part of the *Data* field of the command. The GET-RESPONSE structure details how many more Messages are available for retrieval; and
    c) the GET and GET-RESPONSE pattern repeats until all Messages have been transferred or the GSME decides to stop requesting Messages.

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

10.2.4 TransferData commands sent between GSME and CHF

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

0x03 || Message

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

- the concatenation

  0x02 || (Number of Messages remaining for retrieval after this Message) || (Message addressed to the GSME)

  where it has Messages for the GSME not yet downloaded by the GSME; or
- the concatenation 0x02 || 0x00, where it has no Messages for the GSME to retrieve, the 0x00 representing the number of Messages available to retrieve.

10.3 GSME and GPF interactions

10.3.1 Introduction – informative

The GSME is informed that Remote Party or SME.C.PPMID-GSME Commands are available for it to retrieve via *Tunnel Message Pending* flag on the GPF.

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

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

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

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

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

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

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

10.3.2 GSME data residing on the *ZigBee Gas Mirror Endpoint* – informative

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

10.3.3 GSME data residing on the *ZigBee Gas ESI Endpoint* – informative

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

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

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

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

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

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

10.3.4 GSME Command retrieval and TOM Requirements

10.3.4.1 TOM Commands and Responses

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

- 0x006B (GCS01a Set Tariff and Price on GSME);
- 0x006F (GCS05 Update Prepayment Configurations on GSME) – the GPF shall only process the *Calendar* cluster ZSE commands within the Command;
- 0x0071 (GCS07 Send Message to GSME);
- 0x0015 (CS11 Clear ZigBee Device Event Log) where the Command is addressed to the GSME;
- 0x007C (GCS23 Set CV and Conversion Factor Value(s) on the GSME);
- 0x007E (GCS25 Set Billing Calendar on the GSME);
- 0x00D8 (GCS25a Set Billing Calendar on the GSME – all periodicities);
- 0x0088 (GCS44 Write Contact Details on GSME); or
- 0x00A3 (GCS01b Set Price on GSME).

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

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

The CHF, GPF and GSME shall undertake the processing steps below following receipt of a Remote Party or SME.C.PPMID-GSME Command by the Communications Hub, where that Command is addressed to a GSME on the same SMHAN:

1. the CHF shall buffer the Command and instruct the GPF to set the *Tunnel Message Pending* flag to inform the GSME that the Command is awaiting retrieval. If the Command has been sent as multiple GBT Messages, the GPF *Tunnel Message Pending* flag shall only be set once all GBT Messages making up the Command have been received by the Communications Hub. If not all GBT Messages making up a Command have been received by a Communications Hub within 24 hours of the first GBT Message in that Command being received, then the CHF may discard the GBT Messages that have been received for that command;
2. if 24 hours elapse after setting the GPF *Tunnel Message Pending* flag without the Command being retrieved by the GSME, the CHF may discard the Command. If the CHF discards a Command in this way, it shall notify the GPF and the GPF shall log the event in its Event Log and send an Alert with a GBZ Payload containing an Alert Code 0x819D;

3. when the GSME turns its HAN Interface on, it shall read the *Tunnel Message Pending* flag and retrieve the Command using the *TransferData* command as defined in Section 10.2.3. Each *TransferData* command received by the GSME shall result in the GSME sending a *DefaultResponse* command;
4. the CHF shall process the *DefaultResponse* commands it receives to establish when the Command has successfully been retrieved by the GSME, and shall provide an indication to the GPF accordingly. The GPF shall, when there are no further Commands or GBT Messages pending retrieval by the GSME, clear the *Tunnel Message Pending* flag;
5. if a Command is a TOM Command, the CHF shall retain a copy of the Command contents. For each such Command, the CHF shall start a response timer at the point where it has received *DefaultResponse* command(s) confirming the GSME has successfully retrieved the Command;
6. once a Command is successfully retrieved by the GSME, the GSME shall process the Command in line with the requirements of the GBCS. Note that (1) this processing shall result in the GSME attempting to send a Response to the Command or an Alert that it has received an invalid Command and (2) if sending a Response, the Response shall, as per the GBCS requirements, detail the success or failure of GSME processing for each instruction within the corresponding Command;
7. the GSME shall not, under normal operating conditions, delay sending the Response and shall, where possible, send it before turning its HAN Interface off;
8. on receipt of a Response that is a TOM Response, the CH shall inspect the Response from the GSME. If the Response indicates successful execution of at least one elemental ZCL / ZSE command in the corresponding TOM Command, the CHF shall transfer a copy of the corresponding TOM Command contents and the TOM Response to the GPF, and shall clear the response timer for the Command;
9. on receipt of a TOM Response and the corresponding TOM Command contents, the GPF shall clear any stored copy it has of a TOM Command and then:
   - if the TOM Command is not future dated, process the elemental ZCL / ZSE commands contained within the Command according to the *status* within the Response, updating data it holds accordingly. Once processed by the GPF, the GPF shall make any updated data available over the WAN and over the HAN to the Devices in the GPF's Device Log;
   - if the TOM Command is future dated, store a copy of the TOM Command without updating any data it makes available over the WAN or HAN;
10. if a Response to a TOM Command has not been received by the Communications Hub when the corresponding response timer reaches 6 hours:
    - the CHF may discard its copy of the TOM Command contents, clear the response timer and notify the GPF accordingly; and
    - on receipt of such a notification, the GPF shall log the event in its Event Log and send an Alert with a GBZ Payload containing an Alert Code 0x819E;
11. for clarity, the CHF shall relay all Remote Party Responses received on its HAN Interface through the WAN interface;
12. whenever the CHF receives an Alert detailing activation of a future dated ZCL / ZSE command from within a TOM Command (so an Alert with Alert Code 0x8F66 where the Message Code in the Alert Payload is 0x006B or 0x00A3), the CHF shall pass a copy of that Alert to the GPF;

13. on receipt of such an Alert the GPF shall compare the Originator Counter in the Alert Payload with the Originator Counter of any copy of a TOM Command it holds with the same Message Code as in the Alert Payload, and:
    - if the Originator Counters match, the GPF shall update the data it shares over the HAN and WAN with the elemental ZCL / ZSE command contained within the TOM Command; or
    - if the Originator Counters do not match or the GPF does not hold a TOM Command with this Message Code, the GPF shall send an Alert with Alert Code 0x819E, as 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.

The GSME shall use the *snapshot cause* '*General*' (0x00000001) for the *snapshot* taken.

The GSME shall push the *snapshot* to the GPF using the *PublishSnapshot* command with a *SnapshotPayloadType* and a *SnapshotSub-Payload* populated in line with the requirements of Use Case 'GCS16a Read GSME Daily Read log(s)'. 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*' (0x00000001) 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.

10.4.2.3 Billing Data Log – informative

SMETS defines Billing Data Log as a log capable of storing the following UTC date and time stamped entries:

- twelve entries comprising Tariff TOU Register Matrix, the Consumption Register and Tariff Block Counter Matrix;
- five entries comprising the value of prepayment credits;
- ten entries comprising the value of payment-based debt payments; and
- twelve entries comprising Meter Balance, Emergency Credit Balance, Accumulated Debt Register, Payment Debt Register and Time Debt Registers [1 … 2].

Requirements for each part are detailed separately in the following Sections.

10.4.2.4 Billing Data Log – Tariff TOU Register Matrix, the Consumption Register and Tariff Block Counter Matrix

The GSME shall capture this *snapshot* at the following trigger points:

- End of Billing Cycle (snapshot cause 'End of Billing Period');
- Change of Payment Mode (snapshot cause 'Change of Meter Mode');
- Change of Tariff (snapshot cause 'Change of Tariff Information'); and
- as specified in Section 13.3.5.10 (snapshot cause 'Change of Supplier').

When it next turns on its HAN Interface, the GSME shall push this *snapshot* to the GPF using the *PublishSnapshot* Command with a SnapshotPayloadType and a SnapshotSub-Payload populated in line with the requirements of Use Case 'GCS15b Read GSME Billing Data Log (change of mode / tariff triggered)'.

The GPF shall store the data contained in the *PublishSnapshot* command in the GPF copy of the GSME Billing data Log.

In the event of a communications outage, the GPF shall retrieve missing *snapshots* using the *GetSnapshot* command (and the relevant notification flag) with the UTC start time field populated based on the last received *snapshot* timestamp, if one has been received, or 0x0000 otherwise.

10.4.2.5 Billing Data Log – value of prepayment credits

Upon completion of processing of a valid prepayment top-up, the GSME shall push the latest five prepayment top-ups to the GPF using the *PublishTop Up Log* command.

The GPF shall store the data contained in the *Publish Top Up Log* command in the GPF copy of the GSME Billing data Log.

If there has been a communications outage, the GPF shall use the *Get Top Up Log* command to retrieve all prepayment top-ups that may have been processed during the

communications outage. The GSME shall set the *Date / Time* field of the *Get Top Up Log* command to the current UTC time.

10.4.2.6 Billing Data Log – payment-based debt payments

Upon completion of processing of a valid prepayment top-up where the GSME has made a debt payment using part of that top-up, the GSME shall push details of that debt payment only to the GPF using the *Publish Debt Log* command.

The GPF shall record the details provided in the GPF copy of the GSME Billing Data Log.

In cases of communications outages, the GPF shall request any outstanding payment-based debt payments by use of the *GetDebtRepaymentLog* command (and *GetDebtRepaymentLog* notification flag) with the Debt Type field set to 0x02 (Debt 3).

10.4.2.7 Billing Data Log – Meter Balance, Emergency Credit Balance, Accumulated Debt Register, Payment Debt Register and Time Debt Registers [1 … 2]

The GSME shall capture this snapshot at the following trigger points:

- End of Billing Cycle (snapshot cause bit 1 set: 'End of Billing Period' , as per PublishSnapshot command);
- Change of Payment Mode (snapshot cause bit 14 set: 'Change of Meter Mode');
- Change of Tariff (snapshot cause including set: bit 3 'Change of Tariff Information'); and
- as specified in Section 13.3.5.10 (snapshot cause 'Change of Supplier').

When it next turns on its HAN Interface, the GSME shall push this *snapshot* to the GPF using the *Publish Prepay Snapshot* command.

The GPF shall store the data contained in the *Publish Prepay Snapshot* command in the GPF copy of the GSME Billing Data Log.

In the event of a communications outage, the GPF shall retrieve missing *snapshots* using the *GetPrepaySnapshot* command (and *GetPrepaySnapshot* notification flag) with the UTC start time field populated based on the last received snapshot timestamp, if one has been received.

10.4.2.8 GPF Profile Data Log

The GPF shall create the GPF Profile Data Log from the consumption information pushed by the GSME each half hour.

The GSME shall, on each half hour, record the following information and push to the GPF:

- the *CurrentSummationDelivered* attribute containing total consumption value (with units of m^3);
- the *CurrentDayAlternativeConsumptionDelivered* attribute containing total consumption today (with units of kWh); and
- the *CurrentDayCostConsumptionDelivered* attribute containing total cost of consumption today (with units of Currency Unit);

Upon receipt of the pushed data, the GPF shall calculate the consumption with units of m^3 over the previous half hour by subtracting its previously recorded total consumption value from the total consumption value now sent.

The resulting value shall be stored in the GPF Profile Data Log.

In the event that there are missing values in the GPF Profile Data Log, the GPF shall interrogate the GSME Profile Data Log using the *GetSampledData (SampleID 0x0000)* command and the *GetSampledData* notification flag to retrieve missing values.

10.4.2.9 GPF Daily Gas Consumption Log

The GPF shall create the GPF Daily Gas Consumption Log based on the values pushed from the GSME. The difference between last total consumption value pushed from the GSME each UTC day and the last value pushed in the prior UTC day shall be time stamped and stored in the GPF Daily Gas Consumption Log, so that the values in the log represent consumption in that UTC day.

In the event of communications outages resulting in the final daily value being missed, the GPF shall retrieve the values from the GSME Profile Data Log using the *GetSampledData (SampleID 0x0000)* command and *GetSampledData* notification flag.

10.4.2.10 Historical Attributes

A GSME shall support:

- the *Historical Cost Consumption Information* attribute set, measured in Currency Units; and
- the *Alternative Historical Consumption* attribute set, measured in kWh.

A GPF shall mirror the attribute sets listed above.

As per Section 10.4.2.8, the GSME shall, on each half hour, record the following information and push to the GPF:

- total consumption value (with units of m^3);
- total consumption today (with units of kWh); and
- total cost of consumption today (with units of Currency Unit);

Using the 'total consumption today' value, the GPF shall update the attributes of the mirrored *Alternative Historical Consumption* attribute set.

Using the 'total cost of consumption today' value, the GPF shall update the attributes of the mirrored *Historical Cost Consumption Information* attribute set.

In exception circumstances, the GPF shall request the GSME to push the historical data sets using the '*Push Historical Metering Data Attribute Set* and '*Push Historical Prepayment Data Attribute Set'* notification flags. The GSME shall interpret the '*Push Historical Metering Data Attribute Set'* notification flag as requiring it to push the *Alternative Historical Consumption* attribute set.

10.4.2.11 Other attributes

The GSME, ESME and the SAPC, where it supports such functionality, shall populate the *AccumulatedDebt* attribute in line with the SMETS Accumulated Debt requirements, and all other Devices shall interpret that attribute correspondingly.

The GSME, ESME and the SAPC, where it supports such functionality, shall populate the *Credit Remaining* attribute in line with the SMETS Meter Balance requirements, and all other Devices shall interpret that attribute correspondingly. The GSME, ESME and the SAPC, where it supports such functionality, shall apply functionality related to the *CutOffValue* attribute in line with this interpretation of *Credit Remaining* and the SMETS requirements for Disablement Threshold.

The GPF shall calculate the value of the price in any ZCL *PublishPrice* command it creates using the tariff information it has derived through the TOM and the time from the Communications Hub's clock.

The ESME and the SAPC shall populate the *AuxSwitchNLabel* attribute in line with the SMETS Auxiliary Controller [n] Description requirements, and all other Devices shall interpret that attribute correspondingly.

The *CommodityType* and *MeteringDeviceType* attributes shall be set by devices as follows:

- 'GPF: Metering Device (Gas Mirror Endpoint)': 128 (Mirrored Gas Metering) – *MeteringDeviceType* only;
- 'GSME: Metering Device': 1 (Gas Metering) – *MeteringDeviceType* only;
- 'GPF: Energy Services Interface (Gas ESI Endpoint)': 1 (Gas Metering) – *CommodityType* only;
- 'ESME: Energy Services Interface (Electricity ESI Endpoint)' and not a Polyphase ESME: 0 (Electric Metering);
- 'SAPC: Energy Services Interface (Electricity ESI Endpoint)': 0 (Electric Metering);
- 'ESME: Energy Services Interface (Electricity ESI Endpoint)' and a Polyphase ESME: 15 (Electric Metering Element/Phase 3);
- 'ESME: Energy Services Interface (Twin ESME aggregate ESI Endpoint)': 0 (Electric Metering);
- 'ESME: Energy Services Interface (Twin ESME primary ESI Endpoint)':13 (Electric Metering Element/Phase 1); and
- 'ESME: Energy Services Interface (Twin ESME secondary ESI Endpoint)':14 (Electric Metering Element/Phase 2).

When processing a ZSE *Get Event Log* command or a ZSE *Clear Event Log* command with a Log ID nibble of 0x6 (GSME Event Log) or 0x7 (GSME Security Log), a GPF shall process the command using the relevant GSME Proxy Log copy of the GSME Event or Security Log. Other values of Log ID shall refer to the GPF's own logs.

Where a Device is an ESME which is not a Twin Element ESME, or is an SAPC which supports such functionality, the Device shall populate the *SiteID* attribute with the 13 most significant octets being the Import MPAN and the following 13 octets the Export MPAN, where those terms have their SMETS meanings.

Where an ESME is a Twin Element ESME it shall populate:

- the *SiteID* attribute in the 'ESME: Energy Services Interface (Twin ESME aggregate ESI Endpoint)' with the 13 most significant octets being the Import MPAN on the primary element and the following 13 octets the Export MPAN; and
- the *SiteID* attribute in the 'ESME: Energy Services Interface (Twin ESME secondary ESI Endpoint)' with the most significant 13 octets being the Import MPAN on the secondary element.

Where a ZCL / ZSE command containing *IssuerEventId* and / or *ProviderID* fields is received by a Device as part of a GBZ Remote Party Command, the Device shall undertake no processing in relation to those two fields. For clarity, this means the Device shall not use the contents of those fields for anti replay purposes.

In ZSE *GetSampledData* and *GetSampledDataResponse* commands:

- the *SampleID* field shall be interpreted as:
  - 0x0000 meaning Profile Data Log;
  - 0x0001 meaning Daily Consumption Log; and
  - 0x0002 meaning Network Data Log;

- the *Sample Type* field shall, where the command is to or from an ESME and relates to the Profile Data Log, be interpreted as:
  - 0x00 meaning Consumption / Active Energy Imported;
  - 0x01 meaning Active Energy Exported;
  - 0x02 meaning Reactive Energy Imported; and
  - 0x03 meaning Reactive Energy Exported;
- each of the integers in the *Samples* field shall, when populated by ESME, be the 24 least significant bits of the corresponding value that would be returned when the Profile Data Log is read using a Remote Party Command. Thus, the values returned in *Samples* are in whole Wh / varh; and
- the *SampleRequestInterval* field shall contain 0xFFFF whenever the *SampleID* field is 0x0001.

A GSME shall reject any *PublishPriceMatrix* command that does not contain four *Price* fields.

When processing a *Get Snapshot* or *Get Prepay Snapshot* command, a Device shall return all snapshots where the *Snapshot Cause* in the snapshot matches any of the set bits in the *Snapshot Cause* parameter of the command. When processing a command where *Issuer Calendar ID* has the value 0xFFFFFFFF or 0xFFFFFFFE, a GPF or GSME shall interpret 0xFFFFFFFF as meaning the currently in force Tariff Switching Table calendar and 0xFFFFFFFE as meaning the currently in force Non-Disablement Calendar.

Devices shall support the requirements relevant to their device type detailed in '*Trust Center Swap-Out*' process section of the ZSE specification. Where a Device would normally use the CHF's *Keep Alive Cluster* to periodically check communications with the CHF as part of its '*Trust Center Swap-Out*' support, the Device shall, when first connecting to the HAN:

- Check whether the CHF supports the *Keep Alive Cluster*; and
- Where the CHF does not support the *Keep Alive Cluster*, use an alternative behaviour to allow it to detect a failure to communicate with the CHF.

The GSME shall set the *BillDeliveredTrailingDigit* attribute to the same value as *PriceTrailingDigit* in the *Price* cluster.

In line with the SMETS requirement, the *UnitOfMeasure* parameter in the *PublishTariffInformation* command, sent to a GSME shall be 0x00 (kWh) as per the Message Templates for GCS01a and GCS01b, shall apply to the *Block Threshold N* parameter in the *PublishBlockThresholds* command in such Messages. Contrary to ZSE, the GSME shall undertake the necessary calculation when comparing these thresholds against the *CurrentBlockPeriodConsumptionDelivered* attribute (whose unit of measure in line with SMETS shall be m^3).

Contrary to ZSE, the GPF shall accept any valid UTCTime in the value of the *Implementation Date/Time* parameter in a *Publish Change Of Tenancy* command, in a Command complying with Use Case GCS09.

The GSME shall only action a ZSE *Set Low Credit Warning Level* command it receives at the specified 'From Date Time' specified with that ZSE command.

In GBZ Remote Party Commands containing the *Issuer Tariff ID* parameter, the value of that parameter shall always be 0x00000001.

In GBZ Remote Party Commands containing the *Friendly Credit Calendar ID* parameter, the value of that parameter shall always be 0x00000002.

In responding to GBZ Remote Party Commands requesting information relating to the *Price Cluster*, the GSME & GPF shall provide that information in the GBZ Remote Party Response.

On receipt of a *Credit Adjustment* command with a *Credit Adjustment Type* set to 0x01, the GSME shall reset the prepayment mode Meter Balance, the Emergency Credit Balance and the Accumulated Debt Register.

Where a Device receives a GBZ Command containing a ZSE *Change Payment Mode* command or a ZSE *Get Current Price* command and the Device fails successfully to execute that ZSE command, the Device shall populate the corresponding GBZ Response with a ZSE *Default Response* command in the place of either a ZSE *Change Payment Mode Response* command or a ZSE *Publish Price* command.

For GSME, the value of the *Multiplier* attribute in the *Metering* cluster shall be set to 1 and the value of the *Divisor* attribute in the *Metering* cluster shall be set to 1000.

Where a ZCL / ZSE command containing a ZCL *Date* field is received by a GSME or by a GPF as part of a GBZ Remote Party Command, the Device shall always interpret the value 0xFF in a *Date* sub-field as meaning 'unused' and so having the effect of a wildcard. For clarity, this means the Device shall never interpret the value 0xFF in such *year* sub-fields as meaning the year 2155.

Where a ZCL / ZSE command containing a ZCL *Date* field is received by a Device from a GPF, an SAPC or an ESME, the receiving Device shall always interpret the value 0xFF in a *Date* sub-field as meaning 'unused' and so having the effect of a wildcard. For clarity, this means the Device shall never interpret the value 0xFF in such *year* sub-fields as meaning the year 2155.

ESME, GSME and SAPC, where it supports such functionality, shall set bit 6 of the Prepayment Alarm Status attribute ('*Emergency credit is available to be selected*') to 0b1 only when Emergency Credit is available but not activated (where those terms have their SMETS meanings). Otherwise the attribute shall have the value 0b0.

10.5 Hand Held Terminal (HHT) interactions

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.

This Section 10.5 specifies requirements related to:

- how a connection is made between an HHT and a Communications Hub; and
- 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

ZigBee supports an Inter-PAN mechanism which can be used for communication links between Devices. This mechanism is used to establish a link key between the HHT and the Communications Hub (referred to as the Inter-PAN Link Key (IPLK)), by using ZSE's *CBKE* (for clarity, this IPLK is not used to encrypt any Inter-PAN communication). The Inter-PAN link is then used for HHT to Communication Hub communication:

- the HHT uses the link to send its Entity Identifier and *Install Code* to the Communications Hub as part of a 'CCS01 Add Device to CHF device log' Command;

- the Communications Hub adds these details to the CHF's Device Log (so allowing the HHT to *join* the SMHAN) and confirms this to the HHT using a 'CCS01 Add Device to CHF device log' Response; 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. The IPLK is used as the *pre-configured link key* in this *joining* process.

Both the *Inter-PAN communications* and *joining* to the SMHAN use the *CBKE* mechanism that is defined in ZSE.

Inter-PAN is only to be available for 60 minutes from power on of the Communications Hub. So, if needed, *Inter-PAN* can be enabled by power cycling the Communications Hub.

The *Inter-PAN* mechanism defined by ZSE requires the HHT to specify the Communications Hub that it wishes to link to. There may be multiple Communications Hubs available to the HHT to connect to via *Inter-PAN.*

There are a number of options to provide the HHT with information sufficient to identify uniquely the Communications Hub it is to link to, including:

- the installer manually reading the GPF's Entity Identifier (which is the IEEE address of the Communications Hub's SMHAN radio) printed on the Hub, and confirming / selecting this on the HHT; or
- the installer using a scanner on the HHT to read the GPF's Entity Identifier.

Two illustrative connection scenarios are provided in the following two Sections

10.5.2.1 Illustration 1: Installer manually chooses network – informative

1. the Communications Hub opens Inter-PAN communication for 60 minutes after power on;
2. the HHT is powered on;
3. the HHT performs an active scan using the *Beacon Request* mechanism;
4. the HHT displays the IEEE addresses returned in the *Beacons* from all neighbouring *PAN Coordinators*. Note that the GBCS requires the *Extended PAN ID* initially to be set to the Communications Hub's HAN Interface's IEEE address. This is the same as the GPF Entity Identifier, which is printed on the Communications Hub in line with CHTS. Note that, if the Communications Hub is a replacement for one previously installed, the Extended PAN ID will not be its GPF Entity Identifier, and the installer will need a different way to establish the correct Extended PAN ID to use;
5. the installer (who knows the Consumer's Communications Hub's ZigBee IEEE address as the GPF Entity Identifier is printed on the Communications Hub) picks the desired IEEE address;
6. the HHT initiates *Inter-PAN CBKE* with the Communications Hub;
7. the Communications Hub responds to the *Inter-PAN CBKE*;
8. if *Inter-PAN CBKE* completes successfully, the HHT sends its *Install Code* and Entity Identifier to the Communications Hub in a CCS01 Command, then
9. the Communications Hub adds the HHT to the CHF Device Log and sends a CCS01 Response to the HHT accordingly, and then
10. the HHT *joins* to the SMHAN;
11. otherwise, no link is established.

10.5.2.2 Illustration 2: HHT uses barcode scan – informative

1. the Communications Hub opens Inter-PAN communication for 60 minutes after power on;
2. the HHT is powered on;
3. the HHT optically scans the GPF Entity Identifier printed on the target Communications Hub;
4. the HHT performs an active scan using the *Beacon Request* mechanism;
5. when a Beacon returns an IEEE address equal to the scanned GPF Entity Identifier, the HHT initiates *Inter-PAN communication* with the Communications Hub so identified;
6. the Communications Hub responds to the *Inter-PAN CBKE;*
7. if *Inter-PAN CBKE* completes successfully:
8. the HHT sends its *Install Code* and Entity Identifier to the Communications Hub in a CCS01 Command, then
9. the Communications Hub adds the HHT to the CHF Device Log and sends a CCS01 Response to the HHT accordingly, and then
10. the HHT *joins* to the SMHAN;
11. otherwise, no link is established.

10.5.3 WAN proxy operation

10.5.3.1 Introduction – informative

The HHT has to be capable of holding Remote Party Messages, to which the appropriate Remote Party Message protection has already been applied, and has to be capable of exchanging such Messages.

The Communications Hub must therefore be able to maintain two effective 'WAN' interfaces; the real one via the WAN network interface and a 'logical WAN' via the connection to the HHT.

10.5.3.2 WAN Responses

The Communications Hub shall send any Responses and Alerts through both its WAN interface and the link to the HHT, if present. Whilst this may result in apparent unsolicited Responses at the Remote Party which have to be dealt with, it ensures the earliest possible reconciliation of Commands destined for Smart Metering Equipment.

10.5.3.3 HHT and CHF – Device Requirements

Any Device which provided its Entity Identifier and *Install Code* in a CCS01 Command to the Communications Hub via Inter-PAN shall be treated by the CHF as being of type HHT. For clarity, the Device Type specified in any such CCS01 Command shall be disregarded by the CHF.

As per Section 10.2.2, in interactions between an HHT and a Communications Hub over the SMHAN:

- the HHT shall support the *Tunneling Cluster* as a *Client*; and
- the Communications Hub shall support the *Tunneling Cluster* as a *Server*.

The Communications Hub shall only allow *Inter-PAN communications* for 60 minutes from any power on of the Communications Hub. For clarity, this is the period during which an HHT can establish a connection, not the period of use of any connection.

At power on, a Communications Hub shall remove any Devices of type HHT from the CHF Device Log.

The Communications Hub shall prior to installation, set *nwkExtendedPANId* to be the Entity Identifier of the GPF, which is always the Communications Hub's IEEE address for its HAN Interface.

10.5.3.4 HHT and CHF – establishing communications

Prior to being able to exchange Messages, the HHT and Communications Hub shall undertake the following steps:

1. the HHT shall identify the Communications Hub and initiate the *CBKE* process using *Inter-PAN communications*, as specified in this Section 10.5.3.4;
2. the Communications Hub shall not respond to any such communications if more than 60 minutes has elapsed since the Communications Hub's most recent power on, or if there is a Device of type HHT already in the CHF's Device Log. Otherwise, the Communications Hub shall respond to the *CBKE* request;
3. if *CBKE* does not succeed, processing shall cease. Otherwise, the Communications Hub and HHT shall each store the link key established through CBKE (referred to as the Inter-PAN Link Key (IPLK)) along with a record of the other Device's Entity Identifier, which is its IEEE address. Then processing shall continue from step 4;
4. the HHT shall send a 'CCS01 Add Device to CHF device log' Command, populated with the HHT's Entity Identifier and *Install Code*, to the CHF using the mechanism defined in Section 10.5.3.5;
5. where all parts of the Command are successfully received, the CHF shall first validate that the Entity Identifier in the CCS01 Command matches the Entity Identifier it stored against the IPLK at step 3. If it does not match, processing shall cease. Otherwise the CHF shall process the Command according to the requirements of the 'CCS01 Add Device to CHF device log' Use Case, and shall send any resulting 'CCS01 Add Device to CHF device log' Response to the HHT, via Inter-PAN using the mechanism defined in Section 10.5.3.5. For clarity the *Install Code* in the CCS01 Command shall not be used in the subsequent ZigBee *joining*. Where the CHF sends such a Response, it shall start a timer. When that timer reaches 0xFFFF seconds, the CHF shall remove the HHT from its Device Log, remove the HHT from the SMHAN and close any open tunnels to the HHT;
6. if the 'CCS01 Add Device to CHF device log' Response from the CHF to the HHT states that the HHT has been added to the CHF's Device Log, the HHT may attempt to join the SMHAN. The Devices shall use the IPLK stored at step 3 as the pre-configured link key to secure the joining process. If the *joining* is successful:
   a) the HHT shall send a *RequestTunnel* command to the CHF, with contents as per Section 10.2.2*;*
   b) the CHF shall process the *RequestTunnel* command and send a *RequestTunnelResponse* command in response; and
   c) if *TunnelStatus* in the *RequestTunnelResponse* command is not 0x00 ('*success'*), processing by the HHT shall cease. Otherwise the HHT and CHF may now exchange Messages using the *TransferData* command.

Note that steps 1 to 5 above use *Inter-PAN communications*; the remaining step 6 uses the standard ZigBee SMHAN communications.

Once the HHT has *joined* the SMHAN, any Messages received by the CHF from the HHT in the *Data* parameter payload of a *TransferData* command, shall be forwarded to the relevant Device on the SMHAN as if they were received via the Communications Hub's WAN

interface. If any such Message contains a Business Target ID which is not in the CHF Device Log and is not the CHF ID or the GPF ID, the Communications Hub may silently discard the Message.

Whilst the HHT is in the CHF's Device Log and *joined* to the SMHAN, any Responses received by the CHF from any SMHAN Device shall be provided to the HHT using the *TransferData* command. Such Responses shall also be sent over the Communications Hub's WAN interface, if available.

Note that the requirements of Section 7.2.11(Transfer of Large Remote Party Messages) apply to Messages exchanged once the HHT has joined the SMHAN. Section 7.2.11 requirements do not apply to the CCS01 Command and Response exchanged using the Inter-PAN mechanism.

Once the HHT usage on the SMHAN is complete, the HHT should send a *CloseTunnel* command to the Communications Hub. On receipt of such a *CloseTunnel* command from an HHT, the Communications Hub shall process that command as per the ZSE specification and shall:

- remove the HHT from its Device Log; and
- remove the HHT from the SMHAN.

10.5.3.5 Population of the Inter-PAN ZigBee Frame

A Device may send a Remote Party Message to another Device on the same SM HAN using the ZSE Inter-PAN Frame. In doing so, the sending Device shall:

- Take the 'n' most significant octets of the Remote Party Message to create an 'Inter-PAN Message Fragment (IPMF)', where 'n' is the maximum number such that the aMaxPHYPacketSize (with its IEEE 802.15.4 2003 specification meaning) would not be exceeded in the resulting message over the SM HAN. This IPMF shall have an index of 0x00;
- If there are remaining octets in the Remote Party Message, take the next 'n' most significant octets to create IPMF with an index of one greater than the previous IPMF;
- Repeat step 2 until there are no remaining octets in the Remote Party Message; and
- Set the 'Total Number of IPMF' to be the number of IPMF so created.

A Device receiving a Remote Party Message by this mechanism shall respond to each IPMF received with an 'IPMF Response' constructed according to this Section10.5.3.5. The receiving Device shall re-construct the Remote Party Message in a way that is aligned with the splitting mechanism defined in this Section10.5.3.5.

If a response to an Inter-PAN ZigBee Frame is required by this Section 10.5.3.5 but the receiving Device has not received such a Response after 60 seconds, the receiving Device may treat the transmission of the underlying CCS01 Message as having failed.

When a sending Device is populating the Inter-PAN ZigBee Frame to communicate with a receiving Device using Inter-PAN, the sending Device shall:

- Populate the Inter-PAN ZigBee NWK Header according to the requirements of Table 10.5.3.5a;
- Populate, in the Inter-PAN ZigBee APS Header, the fields specified in Table 10.5.3.5b, according to the requirements of Table 10.5.3.5b; and
- Populate the Inter-PAN ZigBee Payload:
    - According to the requirements of Table 10.5.3.5c when the Device is sending an IPMF in this Frame; or

- According to the requirements of Table 10.5.3.5d when the Device is sending an IPMF Response in this Frame.

Element	Contents	Length (bits)	Note
NWK frame control	See remaining rows in this table.	16 in total	Components as per the following rows in this table
- Frame type (bits 0 -1)	0b11	2	Inter-PAN NWK Frame
- Protocol version (bits 2-5)	0b0010	4	ZigBee Pro
- Remaining sub-fields (bits 6-15)	0b0000000000	10	Unused

Table 10.5.3.5a: Inter-PAN ZigBee NWK Header (two octets total length)

Element	Contents	Length (bits)	Note
APS frame control	See next six rows of this table	8 in total	Components as per the following six rows in this table
- Frame type (bits 0 -1)	0b11	2	Inter-PAN NWK Frame
- Delivery Mode (bits 2-3)	0b00	2	Unicast
- Reserved (bit 4)	0b0	1	Unused
- Security (bit 5)	0b0	1	No security
- ACK request (bit 6)	0b0	1	No ACK requested
- Extended Header Present (bit 7)	0b0	1	No Extended Header Present
Cluster identifier	0xFFFF	16	Manufacturer specific
Profile identifier	0x0109	16	ZSE

Table 10.5.3.5b: Inter-PAN ZigBee APS Header (five octets total length)

Note, the Inter-PAN ZigBee Payloads specified in this Section 10.5.3.5 do not contain a *ZCL Header*.

Element	Contents	Length (bits)	Note
Control byte	0x00	8	Indicating that this frame contains an IPMF
Index of this IPMF	See Note column	8	Shall be interpreted as an unsigned 8 bit integer with a value of 0x00 for the first IPMF, and each subsequent IPMF having a value one higher than the previous

Element	Contents	Length (bits)	Note
			A Device shall only send IPMFs in index order, starting at index 0x00 and shall not send a subsequent IPMF unless it has received a Frame containing an IPMF Response for the previous IPMF with a response status of 0x00
Total number of IPMF	Total number of IPMF	8	Shall be interpreted as an unsigned 8 bit integer. This number shall be calculated by the sending Device and the receiving Device shall not process, in the sense of Section 6, the Remote Party Message until it has successfully received IPMFs for all index values for 0x00 to (Total number of IPMF minus one)
Length of IPMF	Number of octets in the IPMF		Shall be interpreted as an unsigned 8 bit integer
IPMF	The octets making up this IPMF.	Variable	Shall be derived from the Remote Party Message according to this Section 10.5.3.5

Table 10.5.3.5c: Inter-PAN ZigBee Payload contents when sending an IPMF (variable total length)

Element	Contents	Length (bits)	Note
Control byte	0x80	8	Indicating that this frame contains an IPMF Response
Index of the IPMF	Index of the IPMF to which this IPMF Response relates.	8	Shall be interpreted as an unsigned 8 bit integer
Response status	0x00 (meaning successful receipt) or 0x01 (meaning receipt failure)	8	Where a Device receives an IPMF Response containing a response status of 0x00, that Device shall send the next IPMF, or cease transmission, if the response status relates to a final IPMF Where a Device receives an IPMF Response containing a response status of 0x01, that Device may either cease sending IPMFs or may begin re-sending starting from the IPMF with index 0x00

Table 10.5.3.5d: Inter-PAN ZigBee Payload contents when sending an IPMF Response (three octets total length)

10.6 Sub GHz Requirements

In this Section 10.6, 'Duty Cycle' shall mean the percentage of time a Device is transmitting on Sub GHz frequencies. For clarity, any actions taken in relation to managing 'Duty Cycle', including the processing of ZSE *Suspend ZCL Messages* commands, shall only relate to communications with *End Devices* operating on Sub GHz frequencies.

10.6.1 Introduction

This Section 10.6 specifies requirements for Devices which are capable of operating on Sub GHz for their SMHAN operations. Data items defined in this Section 10.6 shall have their defined meaning throughout this Section 10.6. For clarity, the requirements of this Section 10.6 shall therefore apply to SAPC which are capable of operating on Sub GHz for their SMHAN operations.

10.6.2 Communications Hub requirements

10.6.2.1 Operational data items

A Dual Band CH shall maintain the data items specified in Table 10.6.2.1a and shall make those data items available through the DLMS COSEM attributes and objects specified in Table 10.6.2.1a, through the Use Cases specified in Table 20.

Data item	Requirements	OBIS code	Attribute
Operating Sub GHz Channel	The CH shall maintain 'Operating Sub GHz Channel' as a 32 bit bit-string specifying the Sub GHz Channel currently in use on the SMHAN The channel is specified by a channel number within 'page' The bit-string shall follow the same structure as the ZSE Channel Change Attribute. Thus, there shall only be one of bits 0-26 set to 0b1, which identifies the channel number in use, and bits 27-31 shall contain the page number in which the channel number is located (so a binary value equivalent to 28, 29, 30 or 31) Note that, as for 2.4 GHz only CH, there is no requirement on the CH to expose the 2.4 GHz channel in use	0-0:94.44.10.2	2
Sub GHz Channel Log	The CH shall maintain 'Sub GHz Channel Log' as a circular log containing date-time stamped entries of the last 100 Operating Sub GHz Channel values used, where each entry in the Log specifies (1) the date-time operation began on the Sub GHz Channel, (2) the Sub GHz Channel (by way of a 32 bit bit-string which has the same structure as Operating Sub GHz Channel) and (3) a flag indicating the trigger for the change In DLMS COSEM terms, the Log shall be exposed as an 'entry_eventLogEntry8' structure as defined in Section 18.2.1.1. The CHF shall create each log entry according to the requirements of 10.6.2.8 For clarity, the first entry created in this Log shall detail the Sub GHz Channel selected by the CH at initial SMHAN network formation	0-0:94.44.10.3	2
CHF Communication Store – Sub GHz link quality.	The CH shall maintain a store holding, for each Device in the CHF Device Log the quality of the any Sub GHz Communications Link established with the CHF by maintaining the '*AverageRSSI*' as specified in Section 10.6.2.1.1	0-0:30.3.0.255	Part of attribute 4

Table 10.6.2.1a: Dual Band CH operational data item requirements - external

A Dual Band CH shall maintain the internal data items specified in Table 10.6.2.1b. For clarity, there is no requirement to expose these data items either in a Remote Party Response or to other Devices on the SMHAN.

Data item	Requirements
Poor Communications Change Channel Matrix	The CH shall maintain, for each Device which is not a GSME, the UTC date-time at which a Sub GHz Channel Scan was undertaken by the CH due to that Device reporting poor radio conditions to the CH At the point where a Device which is not a GSME connects to the CH and when the CHF successfully actions a 'DBCH05 Request CHF Sub GHz Channel Scan'

Data item	Requirements
	Command, the CH shall set the UTC date-time to the 'Start of Time' with its Section 9.1.6 meaning
Unrequited Lost GSME Searches	The CH shall maintain the 'Unrequited Lost GSME Searches' as the number of Sub GHz Channel Changes the CH has undertaken to try to re-establish GSME communications since the last time it communicated with the GSME This value shall be reset to zero whenever communications with the GSME are re-established
MacRxUcastDeltaSum Matrix	The CH shall maintain, for each Device, the number of unicast messages received by the CH's MAC layer from that Device on any Sub GHz Channel over the last Duty Cycle Measurement Period
Non GSME Poor Communications Thirty Minute History Matrix	The CH shall maintain, for each Sub GHz Non-GSME Device, a record of 'Noisy' for each of the preceding 150 thirty minute periods which saw at least one *Mgmt_NWK_Unsolicited_Enhanced_Update_notify* command being received by the CH from the Device

Table 10.6.2.1b: Dual Band CH operational data item requirements - internal

10.6.2.1.1 Calculation of AverageRSSI

The CH shall:

- Where communications are received from a Device communicating on Sub GHz frequencies with the CH, measure RSSI as specified in the ZigBee Specification section D.10.2.2.2;
- Where communications are received from a Device not communicating on Sub GHz frequencies with the CH, set RSSI to zero; and
- Use the value of RSSI, to calculate AverageRSSI (part of attribute 4 of DLMS COSEM object with OBIS code 0-0:30.3.0.255) over the period between 00:00:01 to 00:00:00 UTC in the UTC day prior to the current day.

For clarity:

- AverageRSSI shall be calculated across all packets received from the Device in question during the relevant 24 hour period; and
- The values needed to calculate AverageRSSI shall persist across CH power loss and reboot.

10.6.2.2 Constant data items

A Dual Band CH shall have the data items in Table 10.6.2.2 set to the values in Table 10.6.2.2 prior to being installed. These values shall not be capable of change without firmware change.

Data item	Requirements	Value
Suspension Period	This constant shall be the number of minutes for which ZCL communications are to be stopped when Duty Cycle controls are enforced on a Device by the CH. Thus, this shall be the value of the *Suspension Period* parameter in the *Suspend ZCL Messages* command, where the CH does not set that parameter to zero	5 (minutes)
Critical-Regulated Duty Cycle Threshold	This Threshold shall be the CH enforced boundary between 'Critical' and 'Regulated' Duty Cycle where those terms have the ZSE 5.14.6 meaning Thus, the DutyCycleRegulatedThreshold parameter in the NLME-SET-INTERFACE.request shall be 250, meaning 2.5%	2.5%

Data item	Requirements	Value
Duty Cycle Measurement Period	In ZigBee, this is referred to as *aDUTYCYCLEMeasurementPeriod*, and so *aDUTYCYCLEMeasurementPeriod* shall be set to this value in the CH	60 (minutes)
ScanDuration	As per the ZigBee Specification and IEEE 802.15.4, ScanDuration determines the period of time that the CH spends scanning each channel as part of a Sub GHz Channel Scan	5

Table 10.6.2.2: Dual Band CH constant data item requirements - internal

10.6.2.3 Configuration data items

Prior to installation, a Dual Band CH shall have attribute 2 of the object with OBIS Code 0-0:94.44.10.0 populated with the DLMS COSEM structure specified in Table 10.6.2.3.

CH shall not update attribute 2 of the object with OBIS Code 0-0:94.44.10.0 except as a result of a 'DBCH04 Set CHF Sub GHz Configuration'. The CHF shall not process any such Command where the Table 10.6.2.3 requirements for Tag, Length and 'Device Validation Requirements on Value' are not met. Where the CH does not process such a Command as a result of one or more of these checks failing, the CH shall not send a 'DBCH04 Set CHF Sub GHz Configuration' Response but shall issue an Alert and create a Security Log entry in line with the requirements of Section 6.2.4.2.

Whenever a CHF has updated attribute 2 of object with OBIS Code 0-0:94.44.10.0 as a result of a 'DBCH04 Set CHF Sub GHz Configuration' Command, it shall:

1. create an entry in the Event Log with Event Code set to 0x8F2A;
2. send a 'DBCH09 Sub GHz Configuration Changed Sub GHz Alert' with:
   - the Message Code set to 0x0113;
   - the Alert Code set to 0x8F2A; and
   - the Use Case Specific Additional Content set to the updated value of attribute 2 of the object with OBIS Code 0-0:94.44.10.0, so providing full details of the structure in Table 10.6.2.3.

Whenever the CH sets attribute 2 of the object with OBIS code 0-0:94.44.10.1 as a result of receiving a 'DBCH05 Request CHF Sub GHz Channel Scan' Command, the CH shall send a 'DBCH05 Request CHF Sub GHz Channel Scan' Response to confirm that the attribute has been set and then shall undertake the processing required by Section 10.6.2.7.

Data item	Requirements	DLMS COSEM element	Tag	Length	Device Validation Requirements on Value
Sub GHz Configuration Settings	This shall be a structure containing 15 values	subGhzConfig::= structure {	0x02	0x0F	
Sub GHz Channel Masks	The CHF shall only scan Sub GHz Channels that are allowed to be used by the Sub GHz Channel Masks values Sub GHz Channel Masks shall be a set of four 32 bit bit-strings, one entry for each				

Data item	Requirements	DLMS COSEM element	Tag	Length	Device Validation Requirements on Value
	of the 4 Sub GHz 'pages' (each page equates to a set of channels that could be used in a specific frequency range) The bit-strings shall follow the ZigBee bitmap format for these items in the *Sub GHz Cluster* For clarity: • in the channel part, 0b1 shall mean 'channel can be used by the CH' and 0b0 shall mean 'channel cannot be used by the CH' • bit numbering is in line with GBCS and so is numbered Least Significant Bit is zero (LSB0)				
Page 28 Mask	This shall be the channel mask that the CH applies for *Page 28*, with its ZSE meaning	page28Mask: bit-string(32),	0x04	0x20	At least two of bits 26-0 must be set to 0b1 AND bits 31-27 must have the value 0b11100
Page 29 Mask	This shall be the channel mask that the CH applies for *Page 29*, with its ZSE meaning	page29Mask: bit-string(32),	0x04	0x20	At least two of bits 26-0 must be set to 0b1 AND bits 31-27 must have the value 0b11101
Page 30 Mask	This shall be the channel mask that the CH applies for *Page 30*, with its ZSE meaning	Page30Mask: bit-string(32),	0x04	0x20	At least two of bits 26-0 must be set to 0b1 AND bits 31-27 must have the value 0b11110
Page 31 Mask	This shall be the channel mask that the CH applies for *Page 31*, with its ZSE meaning	Page31Mask: bit-string(32),	0x04	0x20	bits 31-27 must have the value 0b11111
Normal-Limited Duty Cycle Threshold	This Threshold shall define the boundary between 'Normal' and 'Limited' Duty Cycle	normalLimitedDutyCycleThreshold: long-unsigned,	0x12		Must be between 0x0032 and 0x00C8 inclusive (so 50

Data item	Requirements	DLMS COSEM element	Tag	Length	Device Validation Requirements on Value
	where those terms have the ZSE 5.14.6 meaning. In line with the ZigBee Specification, the data item shall be a 16 bit unsigned integer where 100 shall mean 1% and all other values shall be interpreted accordingly				to 200 meaning 0.5% to 2.0%)
Limited-Critical Duty Cycle Threshold	This Threshold shall define the boundary between 'Limited' and 'Critical' Duty Cycle where those terms have the ZSE 5.14.6 meaning. In line with the ZigBee Specification, the data item shall be a 16 bit unsigned integer where 100 shall mean 1% and all other values shall be interpreted accordingly	limitedCriticalDutyCycleThreshold: long-unsigned,	0x12		Must be between 0x0096 and 0x00FA exclusive (so 150 to 250 meaning 1.5% to 2.5%). Must be greater than nornalLimitedDutyCycleThreshold
Maximum Sub GHz Channel Changes Per Week	The CH shall not undertake more than this number of Channel Changes in any 7*24*60*60 second period except where the one or more additional channel changes results from 'DBCH05 Request CHF Sub GHz Channel Scan' Command(s)	channelChangesPerWeek: unsigned,	0x11		Must be between 0x01 and 0x07
GSME Curfew	Shall be the numbers of hours without GSME communications before the CH determines the GSME to be a 'Lost GSME' When the CH identifies that there is a 'Lost GSME', it shall take actions as defined in Section 10.6.2.6	gsmeCurfew: long-unsigned,	0x12		Must be greater than 0x0001
Channel Quieter Threshold	Shall be the minimum number of decibels by which an alternative channel needs to	quieterDecibelsLimit: unsigned,	0x11		Must be 0x01 or greater

Data item	Requirements	DLMS COSEM element	Tag	Length	Device Validation Requirements on Value
	be quieter, in the conditions defined in 10.6.2.8				
Channel Noisier Threshold	Shall be the maximum number of decibels by which an alternative channel may be noisier, in the conditions defined in 10.6.2.8	noisierDecibelsLimit: unsigned,	0x11		Must be between 0x00 and 0x14 inclusive
Non GSME Poor Communications Percentage Threshold	Shall be an integer value between 100 and 10,000 where 100 equates to 1%	nonGsmePoorCommsPercentage: long-unsigned,	0x12		Must be between 0x0064 and 0x2710 inclusive
Non GSME Poor Communications Thirty Minute Periods Measurement Periods	Shall be the number of 30 minute periods over which assessment of non GSME Device poor communications reports is made by the CH	nonGsmeMeasurementPeriods: unsigned,	0x11		Must be between 0x32 and 0x96 (so between 50 and 150) inclusive
Local CH Noise Measurement Period	The number of trailing minutes over which the CH shall assess its own percentage of retried messages	localChNoiseMeasurementPeriod: long-unsigned,	0x12		Must be greater than 0x003C (so 60 minutes)
Local CH Failure Percentage	Shall be an integer value between 100 and 10,000 where 100 equates to 1%	localChFailurePercentage: long-unsigned,	0x12		Must be between 0x0064 and 0x2710 inclusive
Local CH Retry Percentage	Shall be an integer value between 100 and 10,000 where 100 equates to 1%	localChRetryPercentage: long-unsigned}	0x12		Must be between 0x0064 and 0x2710 inclusive

Table 10.6.2.3: Dual Band CH configuration data item requirements

10.6.2.4 CHF Sub GHz Alerts and corresponding events

Dual Band CH shall be capable of generating all Sub GHz Alerts and corresponding Log Entries, and shall generate the triggering events, and corresponding Alerts and Log Entries in line with the requirements of Table 10.6.2.4 and Table 16.2.

For clarity, in relation to any Sub GHz Alert:

- the Business Target ID shall always be the Entity Identifier in the CHF's {`accessControlBroker, keyAgreement, management`} Trust Anchor Cell;
- the Business Originator ID shall be the CHF's Entity Identifier; and
- the Alert Payload shall always, as a DLMS COSEM based payload, be constructed as per Table 7.2.9c.

Event / Alert Code Meaning	Requirements
Limited Duty Cycle Action Taken	This event shall occur when the CH measurement of Duty Cycle rises above the Normal-Limited Duty Cycle Threshold. When this occurs, the CH shall: 1) identify the Device which has the largest value from the MacRxUcastDeltaSum Matrix and set 'Device ID' accordingly; 2) create an entry in the Event Log with Event Code set to 0x8F20 and otherInfo set to 'Device ID'; 3) send a 'DBCH06 Limited Duty Cycle Action Taken Sub GHz Alert' with: a) the Message Code set to 0x0110; b) the Alert Code set to 0x8F20; and c) the Use Case Specific Additional Content set to the concatenation 0x0908 || 'Device ID'; 4) if 'Device ID' is not that of a GSME, the CH shall send to that Device a *Suspend ZCL Messages* command with the *Suspension Period* parameter set to Suspension Period; and 5) if 'Device ID' is that of a GSME, in the *Suspend ZCL Messages* command response to the next *Get Suspend ZCL Messages Status* command received by the CH from that GSME, the CH shall set the *Suspension Period* parameter to Suspension Period. For clarity, SMHAN communications with the specified Device will not be possible for Suspension Period
Duty Cycle fallen below Normal-Limited Duty Cycle Threshold	This event shall occur when the CH measurement of Duty Cycle falls back below the Normal-Limited Duty Cycle Threshold
Critical Duty Cycle Action Taken	This event shall occur when the CH takes the action in ZSE 5.14.6 point 6 For clarity, SMHAN communications with any Sub GHz End Device except GSME will not be possible for Suspension Period starting at the date-time in the Alert / CHF Event Log Entry. Further, OTA firmware downloads to GSME will pause for this period. No Remote Party Commands will be sent to Sub GHz End Devices during this period (although Alerts and Responses may be received from GSME)
Duty Cycle fallen below Limited-Critical Duty Cycle Threshold	This event shall occur when the CH measurement of Duty Cycle falls back below the Limited-Critical Duty Cycle Threshold

Event / Alert Code Meaning	Requirements
Regulated Duty Cycle Action Taken	This event shall occur when the CH takes the action in ZSE 5.14.6 point 7
Duty Cycle fallen below Critical-Regulated Duty Cycle Threshold	This event shall occur when the CH measurement of Duty Cycle falls back below the Critical-Regulated Duty Cycle Threshold
Sub GHz Channel Changed	This event shall occur when a CH begins operating on a new Sub GHz Channel. For clarity, this includes the Sub GHz Channel selected by the CH on SMHAN network formation The resulting Alert and Log entry shall be created as per the requirements of Section 10.6.2.8 and Section 16.2. Note these events are recorded in the dedicated Sub GHz Channel Log and not in the CHF Event Log
Sub GHz Channel Scan initiated	This event shall occur whenever the CH undertakes the processing of Section 10.6.2.8.
Sub GHz Channel Scan Request Assessment Outcome	This event shall occur when a Sub GHz Channel Scan trigger has been assessed by the CH, as per the requirements of Section 10.6.2.7. The resulting Alert and Event shall record both the nature of the triggering event and the outcome of the assessment checks, so including whether a resulting Sub GHz Channel Scan was triggered
Sub GHz Channel not changed due to Frequency Agility Parameters	This event occurs when a Sub GHz Channel scan has been undertaken but the CH determines not to change the Sub GHz Channel
Three Lost GSME Searches Failed	When 'Unrequited Lost GSME Searches' reaches three and 24 hours has elapsed since the most recent Sub GHz Channel Change, this event shall occur
Sub GHz Configuration Changed	This event shall occur when the CHF updates attribute 2 of object with OBIS Code 0-0:94.44.10.0
Message Discarded Due to Duty Cycle Management	The event shall occur as specified in Section 10.6.2.5.2
No More Sub GHz Device Capacity	The event shall occur when: • a Device is added to the CHF Device Log which is not a GSME, SAPC or HCALCS; • there are already four Devices in the CHF Device Log, which are not HCALCS, SAPC or GSME, that joined the SMHAN on a Sub GHz frequency; and • the Device added then attempts to join the SMHAN on a Sub GHz Frequency. On occurrence of this event, the CH shall: 1) not allow the Device to join the SMHAN on a Sub GHz Frequency; 2) create an entry in the Event Log with Event Code set to 0x8F2D and otherInfo set to 'Device ID' of the Device concerned; and 3) send a 'DBCH11 No More Sub GHz Device Capacity Sub GHz Alert' with:

Event / Alert Code Meaning	Requirements
	a) the Message Code set to 0x0115; b) the Alert Code set to 0x8F2D; and c) the Use Case Specific Additional Content set to the concatenation 0x0908 || 'Device ID'

Table 10.6.2.4: CHF Sub GHz Alerts and related Events

10.6.2.5 CHF functional requirements - general

Dual Band CH shall always form a 'Multi-MAC' network with the meaning of ZSE section 5.14.2.

A Dual Band CH may turn off its Sub GHz HAN interface when either (i) no Device in its CHF Device Log has joined the SMHAN on a Sub GHz frequency or (ii) it has not communicated with any Device in its CHF Device Log on a Sub GHz frequency in the prior trailing 7*24*60*60 seconds. For clarity and in line with ZSE, a Dual Band CH shall always have its Sub GHz HAN interface turned on whenever joining is permitted, with the meaning of ZSE 5.4.1.

10.6.2.5.1 Sub GHz GSME message delivery

Whenever a CH is limiting communications to a Sub GHz GSME pursuant to ZSE 5.14.6, the CH shall buffer any Remote Party Commands or SME.C.PPMID-GSME Messages for the Device until either they can be delivered to the Device or 24 hours has elapsed since receipt by the CH.

If 24 hours have elapsed since receipt of a Message by the CH without successful delivery to the GSME, the CH may discard the Message. If the CHF discards a Remote Party Command in this way, it shall notify the GPF and the GPF shall log the event in its Event Log and send an Alert with a GBZ Payload containing an Alert Code 0x819D.

10.6.2.5.2 Sub GHz Non GSME Device message delivery

Whenever a CH is limiting communications to a Sub GHz Non GSME Device pursuant to ZSE 5.14.6, the CH shall on receipt of any Remote Party Command for that Device:

- set 'Additional Data' to be the following concatenation, with values taken from the Remote Party Message:

  0x09 || 0x19 || value Business Originator ID || value of Business Target ID || value of CRA Flag || value of Originator Counter;
- discard the Remote Party Message;
- create an entry in the Event Log with Event Code set to 0x8F2C and otherInfo set to the value of Business Target ID; and
- send a 'DBCH10 Message Discarded Due to Duty Cycle Management Sub GHz Alert' with:
  - the Message Code set to 0x0114;
  - the Alert Code set to 0x8F2C; and
  - the Use Case Specific Additional Content set to 'Additional Data'.

10.6.2.6 CHF functional requirements – channel scan triggers

Whenever any of the trigger conditions in Table 10.6.2.6 occurs, the CHF shall set the value of scanTrigger according to Table 10.6.2.6 and undertake the processing required by Section 10.6.2.7.

The scanTrigger value of 0xFF shall be reserved.

Trigger	Value of scanTrigger
The CH sends a Response to a 'DBCH05 Request CHF Sub GHz Channel Scan' Command	0x00 (meaning Remote Party Command)
The CH receives a *Mgmt_NWK_Unsolicited_Enhanced_Update_notify* command from the GSME	0x01 (meaning GSME request)
With their CHTS meanings, the difference between the 'CH Date and Time' and the date and time in the 'CHF Communications Store' for the GSME rises above 'GSME Curfew' hours	0x02 (meaning GSME missed its curfew)
The time since the last Sub GHz Channel Change rises above 24 hours AND the CH has had no communications with the GSME in the period since that last Sub GHz Channel Change AND 'Unrequited Lost GSME Searches' is less than 3.	0x03 (meaning GSME missing for the last day)
The CH calculated message failure rate averaged over the prior 'Local CH Noise Measurement Period' minutes exceeds 'Local CH Failure Percentage'. The CH shall, in line with the ZSE specification, calculate the message failure rate as the count of the number of unicast messages transmitted at the MAC layer of its Sub GHz interface that fail to be delivered divided by the total number of such unicast messages transmitted	0x04 (meaning CH detected message failure problems)
The CH calculated message retry rate averaged over the prior 'Local CH Noise Measurement Period' minutes exceeds 'Local CH Retry Percentage'. The CH shall, in line with the ZSE specification, calculate the message retry rate as the count of the number of unicast messages transmitted at the MAC layer of its Sub GHz interface that require retry divided by the total number of such unicast messages transmitted	0x05 (meaning CH detected message retry problems)
The 'Non GSME Poor Communications Thirty Minute History Matrix' for a Device records that more than 'Non GSME Poor Communications Percentage Threshold' percent of the entries in the most recent 'Non GSME Poor Communications Thirty Minute Periods Measurement Periods' thirty minute periods are marked as 'Noisy' AND the 'Poor Communications Change Channel Matrix' records, for this Device, that more than 31*24*60*60 seconds have elapsed since a Sub GHz Channel Scan was undertaken for this scanTrigger for this Device	0x06 (meaning Sub GHz Non-GSME Device request)
The CH is forming an SMHAN. For clarity, this shall only happen when a CH is first installed. It shall not happen on any subsequent power cycling or reboot	0x07 (meaning SM HAN formation)

Table 10.6.2.6: Sub GHz Channel Scan triggers

10.6.2.7 CHF functional requirements – scan trigger assessment

If scanTrigger is not 0x07 (meaning SM HAN formation), the CH shall undertake the checks in Table 10.6.2.7a, in the sequence specified, prior to undertaking any Sub GHz Channel Scan. Should a check fail, the CH shall set the statusCode to the value equating to the failed check in Table 10.6.2.7a and shall not undertake further checks.

If all of the checks are passed or scanTrigger is 0x07 (meaning SM HAN formation), the CH shall set statusCode to 0x00 (meaning scanRequestAccepted).

Having set statusCode, the CH shall then:

1. set scanRequestAssessmentOutcomeAndTrigger to the concatenation:

   0x02020901 || statusCode || 0x0901 || scanTrigger;

2. create an entry in the Event Log with Event Code set to 0x8F28 and otherInfo set to scanRequestAssessmentOutcomeAndTrigger; and

3. send a 'DBCH08 Sub GHz Channel Scan Request Assessment Outcome Sub GHz Alert' with:
   a) the Message Code set to 0x0112;
   b) the Alert Code set to 0x8F28; and
   c) the Use Case Specific Additional Content set to scanRequestAssessmentOutcomeAndTrigger.

The CHF shall then undertake the processing specified in Table 10.6.2.7b dependent on the values of statusCode and scanTrigger. Where that processing requires another scan trigger assessment, the CHF shall set scanTrigger in the subsequent scan to have the same scanTrigger value as this scan.

Sequence	Check	statusCode on failure of Check
1	There must not be a Device of type HHT in the CHF Device Log	0x01 (meaning hhtConnected)
2	The CH's current Duty Cycle measurement must be below the Limited-Critical Duty Cycle Threshold	0x02 (meaning dutyCycleUsageIsTooHigh)
3	The CH must not be currently in a state where joining is permitted, with the meaning of ZSE 5.4.1.	0x03 (meaning joiningIsCurrentlyPermitted)
4	If the Communications Hub has sent at least one *Image Block Response* command to a GSME relating to a currently stored GSME Firmware image and the CH has communicated with the GSME in the last 24 hours, then the CH must have received a corresponding *Upgrade End Request* command from that GSME	0x04 (meaning gsmeOTADistributionUnderway)
5	If the trigger for the Channel Scan was not a 'DBCH05 Request CHF Sub GHz Channel Scan' Command, there must be no entries in the CHF Event Log that: • contain Event Code of 0x8F28 (Sub GHz Channel Scan Request Assessment Outcome); and • contain a statusCode of 0x00 (meaning scanRequestAccepted); and • are timestamped within the prior 24*60*60 seconds. Note this means that no Channel Scans can occur in the 24 hours following SMHAN formation unless triggered by a 'DBCH05 Request CHF Sub GHz Channel Scan'	0x05 (meaning tooManyScansToday)
6	If the trigger for the Channel Scan was a 'DBCH05 Request CHF Sub GHz Channel Scan' Command, there must be no entries in the CHF Event Log that: • contain an Event Code of 0x8F28 (Sub GHz Channel Scan Request Assessment Outcome); and • contain a scanTrigger of 0x00 (meaning Remote Party Command); and • contain a statusCode of 0x00 (meaning scanRequestAccepted); and	0x06 (meaning tooManyCommandsToday)

Sequence	Check	statusCode on failure of Check
	• are timestamped within the prior 24*60*60 seconds	
7	If the trigger for the Channel Scan was not a 'DBCH05 Request CHF Sub GHz Channel Scan' Command, the number of entries in the Sub GHz Channel Log that are timestamped within the prior 7*24*60*60 seconds must be less than the 'Maximum Sub GHz Channel Changes Per Week'	0x07 (meaning tooManyScansThisWeek)

Table 10.6.2.7a: scan trigger assessment checks

statusCode	Required CHF Actions
0x00 (meaning scanRequestAccepted).	The CHF shall undertake the processing specified in Section 10.6.2.8
0x01 (meaning hhtConnected)	When the Device of type of HHT is removed from the CHF Device Log, the CHF shall undertake a scan trigger assessment as specified by this Section 10.6.2.7
0x02 (meaning dutyCycleUsageIsTooHigh)	When the CH's Duty Cycle measurement next falls back below the Limited-Critical Duty Cycle Threshold, the CHF shall undertake a scan trigger assessment as specified by this Section 10.6.2.7
0x03 (meaning joiningIsCurrentlyPermitted)	When the CH changes from the state where joining is permitted to a state where joining is not permitted, with the meaning of ZSE 5.4.1, the CHF shall undertake a scan trigger assessment as specified by this Section 10.6.2.7
0x04 (meaning gsmeOTAUpgradeUnderway)	Whenever the soonest of (1) the CHF sends an *Upgrade End Request Response* to the GSME or (2) the GSME Firmware image the CHF holds is removed or (3) 24 hours have elapsed, the CHF shall undertake a scan trigger assessment as specified by this Section 10.6.2.7
0x05 (meaning tooManyScansToday)	If scanTrigger relates to a GSME and so has the value 0x01, 0x02 or 0x03 then, after 24 hours have elapsed, the CHF shall undertake a scan trigger assessment as specified by this Section 10.6.2.7 If scanTrigger does not relate to a GSME and so does not have the value 0x01, 0x02 or 0x03 then the CHF shall take no further action
0x06 (meaning tooManyCommandsToday)	The CHF shall take no further action
0x07 (meaning tooManyScansThisWeek)	If scanTrigger relates to a GSME and so has the value 0x01, 0x02 or 0x03 then, after 24 hours have elapsed, the CHF shall undertake a scan trigger assessment as specified by this Section 10.6.2.7 If scanTrigger does not relate to a GSME and so does not have the value 0x01, 0x02 or 0x03 then the CHF shall take no further action

Table 10.6.2.7b: Post scan trigger assessment actions

10.6.2.8 CHF functional requirements – Sub GHz Channel Scan actions

The CH shall complete a Sub GHz Channel Scan.

The CHF shall then set 'Alternative Sub GHz Channel' to:

- The channel which, of the Sub GHz Channels scanned, has the lowest noise, if this lowest noise channel is different than the CH Operating Sub GHz Channel or if there is no existing channel recorded in the CH Operating Sub GHz Channel; or
- The channel which, of the Sub GHz Channels scanned, has the second lowest noise, if the lowest noise channel is the CH Operating Sub GHz Channel.

The CHF shall then set 'Relative Noise' to be the SMHAN radio noise difference in decibels between the CH Operating Sub GHz Channel and the 'Alternative Sub GHz Channel', or to be zero if there is no existing channel recorded in the CH Operating Sub GHz Channel.

The CHF shall then determine whether to change Sub GHz Channel according to Table 10.6.2.8, the value of scanTrigger and the 'Relative Noise'.

The CHF shall comply with the ZSE requirements whenever undertaking a channel change. Additionally, having set the *Channel Change Pending* Functional Notification flag, the CH shall wait until the soonest of either:

- The GSME has read the Channel Change attribute; or
- 60 minutes have elapsed

before changing its Operating Sub GHz Channel.

Where the CHF has changed the Operating Sub GHz Channel, the CHF shall:

1. create an entry in the Sub GHz Channel Log with Event Code set to 0x8F26 and otherInfo set to the concatenation (value of Operating Sub GHz Channel) || 0x0901 || scanTrigger; and
2. send a 'DBCH07 Sub GHz Sub GHz Channel Changed Sub GHz Alert' with:
   a) the Message Code set to 0x0111;
   b) the Alert Code set to 0x8F26; and
   c) the Use Case Specific Additional Content set to 0x02020420 || (value of Operating Sub GHz Channel) || 0x0901 || scanTrigger.

Value of scanTrigger	Actions
0x00 (meaning Remote Party Command)	If 'Relative Noise' is at least 'Channel Quieter Threshold' decibels better, the CHF shall change its Operating Sub GHz Channel to the 'Alternative Sub GHz Channel'
0x01 (meaning GSME request)	If 'Relative Noise' is no more than 'Channel Noisier Threshold' decibels worse, the CHF shall change its Operating Sub GHz Channel to the 'Alternative Sub GHz Channel'
0x02 (meaning GSME missed its curfew)	The CHF shall change its Operating Sub GHz Channel to the 'Alternative Sub GHz Channel'
0x03 (meaning GSME missing for the last day)	The CHF shall change its Operating Sub GHz Channel to the 'Alternative Sub GHz Channel'
0x04 (meaning CH detected message failure problems)	If 'Relative Noise' is at least 'Channel Quieter Threshold' decibels better, the CHF shall change its Operating Sub GHz Channel to the 'Alternative Sub GHz Channel'
0x05 (meaning CH detected message retry problems)	If 'Relative Noise' is at least 'Channel Quieter Threshold' decibels better, the CHF shall change its Operating Sub GHz Channel to the 'Alternative Sub GHz Channel'
0x06 (meaning Sub GHz Non-GSME Device request)	If 'Relative Noise' is no more than 'Channel Noisier Threshold' decibels worse, the CHF shall change its Operating Sub GHz Channel to the 'Alternative Sub GHz Channel'
0x07 (meaning SM HAN formation)	The CHF shall set its Operating Sub GHz Channel to the 'Alternative Sub GHz Channel'

Table 10.6.2.8: Sub GHz Channel Scan triggers

10.6.3 Sub GHz End Devices - functional requirements

A Sub GHz End Device shall:

- not act as a ZigBee router when operating on a Sub GHz Channel;
- where the Device can also support 2.4 GHz operation:
  - on first connecting to a ZigBee network, attempt to establish network communication in the 2.4GHz band. Only where communications are not of sufficient quality, shall the Device attempt to establish network communications in the Sub GHz band; and
  - having connected to a ZigBee network at either 2.4 GHz or at Sub GHz, not attempt to change to the other of 2.4 GHz or Sub GHz except when undertaking a *Trust Centre re-join*, with its ZSE meaning; and
- not use the *Mgmt_NWK_Unsolicited_Enhanced_Update_notify* command to notify the CH of problems with its communications link more frequently than once in any 30-minute period.

10.6.4 Sub GHz GSME - functional requirements

Sub GHz GSME shall wait at least 2 hours from detecting SMHAN interference before indicating that the interference is continuing by way of sending a *Mgmt_NWK_Unsolicited_Enhanced_Update_notify* command to the CH.

When operating on Sub GHz, GSME shall, on each wake up, check the *Functional Notification Flags* for bits 25 (*Energy Scan Pending*) and 26 (*Channel Change Pending*).

If either bit is set (so has a value 0b1) then the GSME shall attempt to retrieve any Commands buffered for it on the CH before turning off its SMHAN radio. For clarity and in line with Section 10.6.2.8, the GSME should attempt such retrieval before reading the CHF *Channel Change* attribute, if it is to maximise the likelihood of Command retrieval before it turns off its SMHAN radio.

If bit 25 is set, the GSME shall disable the SMETS User Interface Commands '4.5.2.4 Check for HAN Interface Commands' and '4.5.2.8 Find Smart Metering Home Area Network and Re-establish Communications Links' until it next turns on its SMHAN radio.

Note that CH may change Sub GHz Channel once every 24 hours to attempt to communicate with a lost GSME. On each such change, the CH will undertake a Sub GHz Channel Scan, meaning that it cannot communicate with a GSME for a period of time. GSME should factor both the 24 hour period and the associated Sub GHz Channel Scan in to their attempts to re-establish lost communications with the CH.

10.6.5 Sub GHz Available Channels

The Sub GHz Available Channels shall be as defined in the ZigBee Specification at section D.10.3.

10.7 SMHAN management

10.7.1 Requests to leave the SMHAN

Where a Device is not a ZigBee *end device* or a Communications Hub, the *nwkLeaveRequestAllowed* attribute of the *NIB (Network layer information base)* shall be set to *FALSE*. This means that, in line with the ZigBee specification, *Leave* commands received by such Devices shall be immediately and silently discarded and such Devices shall not leave the SMHAN on receipt of such *Leave* commands.

When a Device, which is a ZigBee *end device*, has received and executed a ZigBee *Leave* command with a value in the *Rejoin Sub-Field* of 0, the Device shall attempt to re-establish SMHAN communications using the ZSE 5.4.2 mechanisms.

10.7.2 Network Key Change

In line with the 'ZigBee PRO/2007 Layer PICS and Stack Profiles' requirement, Devices shall be capable of maintaining a minimum of two *Network Keys*.

Whenever a CHF adds an Entity Identifier to, or removes an Entity Identifier from, its CHF Device Log, where the type of the Device to which that Entity Identifier relates is other than HHT or GPF, the CHF shall set a 'CHF Device Log Changed Today' flag.

For clarity and in accordance with Section 10.5.3.3, any Device which provided its Entity Identifier and *Install Code* in a CCS01 Command to the Communications Hub via *Inter-PAN* shall be treated by the CHF as being of type HHT.

Once each day, at a time between 02:00 and 04:00 UTC, the CHF shall check whether the 'CHF Device Log Changed Today' flag is set and, if it is set, the CHF shall take the following steps:

1. generate a new *Network Key*;
2. set, via the GPF, the *Network Key Active* flag in the *FunctionalNotificationFlags* and then wait for the soonest of (1) 60 minutes or (2) the GSME requesting the new *Network Key*;
3. for each Device in the CHF Device Log which has an associated *TC Link Key*, unicast the new *Network Key* to the Device by issuing the *APSME-TRANSPORT-KEY.request* primitive with the *DestAddress* parameter set to the address of the Device and the communication to the Device being protected by the associated *TC Link Key*. The *TransportKeyData* sub-parameters shall be set as follows:
   a) the *KeySeqNumber* sub-parameter shall be set to the sequence count value for the new *Network Key*;
   b) The *NetworkKey* sub-parameter shall be set to the new *Network Key*; and
   c) The *UseParent* sub-parameter shall be set to *FALSE*;
4. Send to all Devices in the CHF Device Log the *APSMESWITCH-KEY.request* primitive with the *KeySeqNumber* parameter set to the sequence count value for the new *Network Key*; and
5. unset the 'CHF Device Log Changed Today' flag.

Should any Device not receive the new *Network Key* and the instruction to switch to it, that Device will discover a communication failure. In such circumstances, the Device shall use the ZSE 5.4.2 ('Re-joining a Secured Network') mechanisms to re-establish communications.

For clarity, the CHF shall not send the new *Network Key* to either:

- devices which have Entity Identifiers in the CHF Device Log but which do not have a *TC Link Key* established; or
- devices which do not have an Entity Identifier in the CHF Device Log.

10.7.3 Network Key related requirements

In this Section, all terms in italics shall have their ZSE or ZigBee Specification meaning and ZS shall mean the ZigBee Specification.

When a Device, which is not a CH, receives a new *Network Key*, the Device shall only store that *Network Key* where either:

1. the Device does not currently hold any *Network Key* (so meaning it is being installed); or
2. the Device receives the new *Network Key* encrypted with a hash of its *Trust Center Link Key* (so meaning that a *Trust Center Swapout* is in progress); or

3. the Device receives the new *Network Key* encrypted with its *Trust Center Link Key* (and potentially with an existing *Network Key*) and either:
   a) the value of *KeySeqNumber* for the new *Network Key* is greater than the value of the Device's *nwkActiveKeySeqNumber*; or
   b) the Device's *nwkActiveKeySeqNumber* is greater than 127 and the value of *KeySeqNumber* for the new *Network Key* is not greater than (*nwkActiveKeySeqNumber* + 128) modulo 256.

Where a Device, which is not a CH, stores a new *Network Key*, it shall switch to using that new *Network Key* for outgoing messages where either:

- it does not hold any other *Network Key*;
- it received the new *Network Key* encrypted only with a hash of its *Trust Center Link Key*; or
- it receives a message validly encrypted with the new *Network Key*.

Where a Device stores a new *Network Key* and that storage leads to the Device needing to remove details related to an old *Network Key*, the Device shall remove the *Network Key* that it received furthest back in time, and remove the *nwkSecurityMaterialSet* details associated with that key. Note that, in cases (2.) and (3.b) above, that key would likely <u>not</u> be the one with the lowest *KeySeqNumber*.

Where a Device, which is not a CH, receives a *switch-key* command requesting that it switches to using a new *Network Key*, the Device shall only take action in response to that command where either:

1. the value of the '*sequence number*' parameter in the *switch-key* command is greater than the value of the Device's *nwkActiveKeySeqNumber*; or
2. the Device's *nwkActiveKeySeqNumber* is greater than 127 and the value of the '*sequence number*' parameter in the *switch-key* command is not greater than (*nwkActiveKeySeqNumber* + 128) modulo 256.

Where a Device, which is not a CH, switches to using a new *Network Key*, the Device shall:

- in line with ZS 4.3.4 (and contrary to ZS 4.6.3.4.2), only set the associated *OutgoingFrameCounter* to zero if *OutgoingFrameCounter* is currently greater than 0x80000000; and
- ensure that, in the *IncomingFrameCounterSet* within the *nwkSecurityMaterialSet* for this new *Network Key*:
  - For a Device which is not an End Device, any SenderAddress is an identifier for a Device that is in the Device's nwkNeighborTable;
  - For a Device which is an End Device, the only SenderAddress is the identifier for Device's current parent Device; and
  - In line with ZS 4.6.3.4.2, all IncomingFrameCounters are set to zero.

A Device shall:

- only increment the value of the *IncomingFrameCounter* for the sending Device as a result of processing incoming messages from the sending Device which are secured with the *Network Key* the receiving Device is currently using for outgoing messages; and
- whenever it removes a Device from its *nwkNeighborTable*, also remove that Device's details from the *IncomingFrameCounterSet* within the *nwkSecurityMaterialSet* for the *Network Key* it is currently using to secure outgoing messages.

For the purposes of aging out entries from the *nwkNeighborTable*, a Device shall, where it is a *Router*:

- only use the ZS table 3-58 specified default values for *nwkRouterAgeLimit* and *nwkLinkStatusPeriod*, so 3 and 15 seconds respectively;
- set bit 0 of *nwkParentInformation* to 0b0, and so bit 1 to 0b1, meaning that *End Devices* need to send *End Device Timeout Request* commands as a unicast to refresh the *keepalive timer*;
- Only refresh the *keepalive timer* when the *Network Key* used to secure such *End Device Timeout Request* commands is that currently in use by the Device for its outgoing messages; and
- have the *nwkEndDeviceTimeoutDefault* set to the default 8 (so meaning 256 minutes) in line with ZS table 3-58 and not change the value of a Device's *keepalive* timeout where it receives an *End Device Timeout Request* command with a *Requested Timeout Enumeration Value* greater than 10 (so meaning greater than 1,024 minutes).

When a Device has chosen a network to join, it shall remove *Neighbor table entries* corresponding to Devices that are not members of the chosen network.

Where a Device is an *End Device*, the Device shall not send an *End Device Timeout Request* command with a *Requested Timeout Enumeration Value* greater than 10 (so meaning greater than 1,024 minutes).

CH shall not send any new *Network Key* encrypted only with an existing *Network Key*.

Where a CH creates a switch-key command, it shall treat the reference in ZS 4.4.6.1.3 to ZS 4.4.9.6 (which does not exist) as a reference to ZS 4.4.10.5.

A CH shall not attempt to remove Devices from the network using the *APSME-REMOVE-DEVICE* primitives, and so shall not send *Remove Device* commands to remove Devices from the network. A CH may send *Remove Device* commands only as required by ZSE 5.4.2.2. For clarity, this means that the values in the *ParentAddress* and *TargetAddress* parameters of such an *APSME-REMOVE-DEVICE* request shall never be the same.

10.8 CH routing of Remote Party and SME.C.PPMID-GSME Messages

10.8.1 Defined Terms

The following terms used in this Section 10.8 shall have the meanings defined in this Table 10.8.1.

Defined Term	Meaning
Message Header Structure	Shall, in relation to the Message being processed, be any one of: • a Grouping Header, as specified in Section 7.2.7; • a MAC Header, as specified in Section 7.2.5; • a Message Routing Header, as specified in Section 7.2.11.5; and a GBT Header, as specified in Section 7.2.11.6
Non-Validated Components	Shall be: • in a Grouping Header or a Message Router Heading, any of: ○ The value for Originator Counter in the transaction-id;

Defined Term	Meaning
	○ The value for Business Originator ID in the originator-system-title; ○ Any value in the date-time, where such a value is present; and ○ Any value in other-information; and ○ any values in any parts of a Message that are not within a Message Header Structure.

Table 10.8.1: Meanings of Defined Terms

10.8.2 CH Routing of Remote Party Commands and SME.C.PPMID-GSME and Alerts

Whenever a CH receives either:

- a Remote Party Message via its WAN interface; or
- a Remote Party Message in the Data parameter payload of a TransferData command which is from an HHT; or
- an SME.C.PPMID-GSME Message in the Data parameter payload of a TransferData command from a Device, which is in its CHF Device Log,

the CH shall:

- process the Message Header Structure(s) in that Message sufficiently to identify the target Device's Entity Identifier; and
- where the identified Device is in the CHF Device log and is not an HHT, GPF or CHF, attempt to deliver that Message to the identified Device.

In undertaking such processing, the CH shall <u>not</u> validate any of the 'Non-Validated Components' as a pre-requisite to attempting to deliver the Message to the identified Device.

Attempts at delivery of Messages shall be subject to the provisions of Sections 10.2.2, 10.3.4.2 and 10.6.2.5.1.

10.8.3 CH Routing of Remote Party Responses and Alerts

Whenever a CH receives content in the Data parameter payload of a *TransferData* command which is not from an HHT, the CH shall process that content including any potential Message Header Structure(s) in it, so as to decide whether it contains a Remote Party Message. Where the CH determines that the content contains a Remote Party Message, the CH shall attempt to deliver that Message via its WAN interface and, as required by Section 10.5.3.2, by sending a *TransferData* command containing the Message to any HHT in its CHF Device Log.

In undertaking such processing, the CH shall <u>not</u> validate any of the Non-Validated Components, or the value for Business Target ID in the recipient-system-title, as a pre-requisite to attempting to deliver the Message.

11 Downloading firmware images to Devices

Requirements and statements in this Section 11, and in Use Cases 'CS05b Distribute Firmware to ESME / GSME' and 'CS06 Activate Firmware', shall apply to an SAPC as if it were an ESME.

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 firmware update process being separated into two stages for ESME, GSME, HCALCS and Communications Hubs:

- 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.

For PPMID the firmware image is delivered to the Communications Hub and transferred to the target Device. Upon successful transfer of the firmware image the Communications Hub instructs the PPMID to activate the firmware 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.

In line with the ZigBee OTA specification at section 5.1, the contents of Manufacturer Images sent to Devices are manufacturer defined. Thus, a particular Manufacturer Image may consist of whatever the manufacturer requires to achieve the necessary update which could be a full image or just a patch to application code or any other manufacturer specified content.

Therefore the steps taken by a Device when it activates the contents of a particular Manufacturer Image are manufacturer specific and specified in the release note for that Manufacturer Image. Thus, activation of a Manufacturer Image means the set of manufacturer specified steps the Device takes in processing the content of the image and terms such as 'activation', 'activate', 'activated', 'installed' etc. in the GBCS and the Technical Specifications, are meant in this sense.

Therefore, when a Device reports its 'installed' or 'active' firmware version it shall respond with either:

- the firmware version for its mostly recently successfully activated Manufacturer Image (with the value in the corresponding OTA Header and so the value in the Central Products List (CPL) with its SEC meaning); or
- if it has never successfully activated a new Manufacturer Image, the firmware version it held on installation (which again would correspond to the CPL).

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 09-5264r23.

For ESME, GSME, HCALCS and PPMID firmware image distribution, the ZigBee Over-The-Air (OTA) mechanisms shall be used for transport of the image over the HAN. The ESME / GSME / HCALCS / PPMID 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.

Every Communications Hub shall be configured to act as the single OTA Server on its HAN.

ESME, GSME, HCALCS and PPMID shall be configured to act as an OTA Client. The ESME, HCALCS and PPMID shall use the '*Image Notify*[24] *Command* sent by the OTA Server to inform it that a new firmware image is available. The GSME shall use the notification flags mechanism whereby a flag shall be set by the OTA Server to inform it that a new firmware image is available when requested.

The Communications Hub shall:

- as required by CHTS, have the capability to store one GSME OTA Upgrade Image and one ESME OTA Upgrade Image;~~ and~~
- overwrite an ESME / GSME image with a subsequently delivered image for the same Device type unless:
  - the subsequently delivered image has Force Replace = 0x00; and
  - the Communications Hub has sent at least one *Image Block Response Command* relating to the already stored image but has not received a corresponding *Upgrade End Request Command*[25].

  In such circumstances the Communications Hub shall not overwrite the currently stored image~~.~~; and
- overwrite an HCALCS image with a subsequently delivered HCALCS image regardless of the setting of Force Replace.

Formatted: List Bullet, Indent: Left: 1.25 cm

Formatted: List Bullet

[24] See section 6.10.3 of ZigBee Document 09-5264r23
[25] As defined in section 6.10 of ZigBee Document 09-5264r23

The Communications Hub shall make the GSME image available for a minimum of fourteen (14) days and, after this period, replace the GSME image with an image for the PPMID / HCALCS, if one becomes available. If the transfer of a GSME image to the GSME is in progress, the Communications Hub shall only replace this GSME image with an image for the PPMID / HCALCS once the GSME image transfer has completed.

~~If the transfer of the GSME image is interrupted for fourteen (14) days, the CH shall discard the GSME image.~~If the Communications Hub has sent at least one Image Block Response Command relating to the stored GSME image but has not received a corresponding Upgrade End Request Command within fourteen (14) days, the CH shall discard the GSME image.

The Communications Hub shall make the PPMID / HCALCS image available for fourteen (14) days unless a new PPMID / HCALCS / GSME image is available.

If an ESME / PPMID / HCALCS / GSME Upgrade Image is discarded or replaced prior to having been successfully transported over the HAN, the Communications Hub shall send an Alert for each target Device Entity Identifier associated with the Upgrade Image File with the Alert Code 0x8F89 as specified in Section 11.7 by setting `firmwareVersion` to the Upgrade Image File version and `transferResponseCode` to `imageDiscarded`.

Whenever the Communications Hub's OTA Server issues an *Upgrade End Response Command* to a GSME, ESME or HCALCS pursuant to this GBCS, the *UpgradeTime* parameter shall have the value 0xFFFFFFFF[26].

Whenever the Communications Hub's OTA Server issues an *Upgrade End Response Command* to a PPMID pursuant to this GBCS, the *UpgradeTime* parameter shall have the value 0x00000000 and the PPMID OTA Client shall activate the Firmware.

The OTA Server shall not issue *Image Block Response Commands* with WAIT_FOR_DATA status except where the OTA Server is communicating over a Sub GHz Channel and it is issuing such commands to manage Duty Cycle (with its ZSE meaning).

Contrary to section 6.13 of ZigBee Document 09-5264r23, the OTA Client shall not activate any ESME, GSME or HCALCS Firmware ~~for an ESME, GSME or HCALCS~~ except as specified in Use Case CS06.

~~Whenever the Communications Hub's OTA Server issues an *Upgrade End Response Command* to a PPMID pursuant to this GBCS, the *UpgradeTime* parameter shall have the value 0x00000000 and the PPMID OTA Client shall activate the Firmware.~~

Formatted: Font: Italic

Formatted: Font: Italic

11.2.2 Construction of Upgrade Image

For an ESME, GSME or HCALCS firmware image, the Authorising Remote Party shall be the Supplier for the target Device.

For a Communications Hub firmware image, the Authorising Remote Party shall be the WAN Provider for the target Device.

For an ESME, GSME, HCALCS or Communications Hub the Upgrade Image shall be the concatenation:

Manufacturer Image || Force Replace || 0x40 || Authorising Remote Party Signature

where:

- Manufacturer Image shall contain the firmware image the Device is to apply and any manufacturer specific data needed. For clarity, the GBCS shall not constrain the structure or contents of Manufacturer Image;

[26] As defined in sections 6.10.10 and 6.8.4 of ZigBee Document 09-5264r23

- Force Replace shall be a single octet where Force Replace = 0x00 shall mean do not force the replacement of the currently stored image; and
- Authorising Remote Party Signature shall be calculated across the Manufacturer Image using the Authorising Remote Party's Private Digital Signing Key.

For a PPMID firmware image, the Upgrade Image shall be the Manufacturer Image where:

- Manufacturer Image shall contain the firmware image the PPMID is to apply and any manufacturer specific data needed. For clarity, the GBCS shall not constrain the structure or contents of the PPMID Manufacturer Image.

11.2.3 Construction of OTA Upgrade Image

OTA Upgrade Image shall be the concatenation:

OTA Header || Upgrade Image

where OTA Header shall be populated according to Table 11.2.3. For clarity, there shall be 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	Detailing 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	

Table 11.2.3: Population of the OTA Header

The OTA Header shall uniquely identify a firmware image.

11.2.4 Construction of Manufacturer Image Hash

Manufacturer Image Hash shall be a Hash calculated across the whole Manufacturer Image file that is provided to the Authorising Remote Party or the Remote Party in case of the PPMID.

11.2.5 Verification of the authenticity of the Upgrade Image

The Device shall verify Upgrade Image by verifying the Authorising Remote Party Signature using Manufacturer Image and the Authorising Remote Party's Public Key. For clarity, this shall be the only ECDSA verification required by the GBCS and this is not the ZSE ECDSA Signature sub-element.

For an ESME, GSME or HCALCS receiving an Upgrade Image, the Authorising Remote Party's Public Key shall be that held by the Device in the {`supplier, digitalSignature, management`} Trust Anchor Cell.

For a Communications Hub receiving an Upgrade Image, the Authorising Remote Party's Public Key shall be that held by the Device in the {`wanProvider, digitalSignature, management`} Trust Anchor Cell.

For a PPMID receiving an Upgrade Image the verification of the authenticity of the Upgrade Image in this section shall not apply.

11.2.6 Construction of Firmware Distribution Receipt Alert

If the Device is an ESME or HCALCS, the 'Alert Payload' fields shall be populated according to Section 7.2.9.

If the Device is a GSME, the 'Alert Payload' fields shall be populated according to Section 7.2.10.

In the above cases, the Device shall:

- populate the Use Case Specific Additional Content with the concatenation 0x0920 || the calculated Manufacturer Image Hash
- populate the Alert Code field with 0x8F1C (failure), or 0x8F72 (success).

If the Device is a PPMID, no Firmware Distribution Alert shall be sent.

11.2.7 Activation of firmware images

The Activate Firmware Command shall be of type SME.C.C.

A Device other than the PPMID receiving such a Command shall undertake the verifications required of a SME.C.C Command.

If all such SME.C.C verifications succeed, the Device shall then calculate Manufacturer Image Hash over the Manufacturer Image it holds and compare that with the Manufacturer Image Hash specified in the Activate Firmware Command (see Use Case CS06 in Section 11.5 for details of the Activate Firmware Command Payload construction).

If the two Hashes match, the Device shall attempt to activate the firmware image.

If the two Hashes do not match, the Device shall not attempt to activate the firmware image.

The Device shall issue a relevant Activate Firmware Response detailing the success or failure (see Use Case CS06 in Section 11.5 for details of the Activate Firmware Command Payload construction).

See Section 11.6 for the activation~~handling~~ of Upgrade Images for the PPMID.

11.3 CS05a Distribute Firmware to Communications Hub

This Use Case covers the distribution of an Upgrade Image that is intended for a 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
Service Reference	N/A (this Command does not have a Service Reference for CH)
Valid Target Device(s)	Communications Hub
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]	WAN Provider
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	WAN Provider Specific

Table 11.3: Use Case Cross References for CS05a Distribute Firmware to Communications Hub

11.3.1 Pre-conditions

None.

11.3.2 Detailed Steps

The Upgrade Image shall be constructed according to Section 11.2.2.

The Upgrade Image shall be transported to the Communications Hub.

The Communications Hub shall verify the Upgrade Image according to Section 11.2.5, verify the Upgrade Image is suitable for this Communications Hub, and discard the Upgrade Image where it fails to verify.

11.4 CS05b Distribute Firmware to ESME / GSME / HCALCS

This Use Case covers the distribution of an OTA Upgrade Image that is intended for a GSME, ESME or HCALCS to that GSME, ESME or HCALCS.

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Command and Response

Cross Reference	Value
Message Type Category	None – this is a Variant Message
Capable of future dated invocation?	No
Protection Against Replay Required?	No
Service Reference	11.1
Valid Target Device(s)	ESME / GSME / HCALCS
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	WAN Provider Specific to Communications Hub; ZigBee OTA from Communications Hub to ESME / GSME / HCALCS

Table 11.4: Use Case Cross References for CS05b Distribute Firmware to ESME / GSME / HCALCS

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, GSME and HCALCS shall use the ZCL *Image Block Request* and *Image Block Response* commands to retrieve available OTA images.

ESME, GSME and HCALCS 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, HCALCS and GSME:

- for ESME and HCALCS, 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*.

For each ESME / GSME / HCALCS ~~target~~ Device Entity Identifier associated with the OTA Upgrade Image ~~version~~, the Communications Hub shall

1. verify whether the ESME / GSME / HCALCS *Hardware Version* is present in the *Query Next Image Request Command;*

- if the *Hardware Version* is present and does not fall between~~match~~ the *minimum* and *maximum Hardware Version* in the OTA Header the Communications Hub shall set `transferResponseCode` to ~~`hardware`~~`hardwareVersionMismatch` as specified in Section 11.7 and process from step 5;
- if the *Hardware Version* is not present, or is present and falls between~~matches~~ the *minimum* and *maximum Hardware Version* in the OTA Header, the Communications Hub shall make the OTA Upgrade Image available to the ESME / GSME / HCALCS in the *Query Next Image Response Command*;

2. The ESME / GSME / HCALCS 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* specification[27];
3. upon the reception of the *Upgrade End Request Command* from the ESME / GSME / HCALCS, the Communications Hub shall;
   - set `transferResponseCode` to `fileTransferSuccess` if the transfer of the OTA Upgrade~~Manufacturer~~ Image has completed successfully and process from step 4;
   - set `transferResponseCode` to `fileTransferFailure` if the transfer of the OTA Upgrade~~Manufacturer~~ Image has failed~~ and all transfer retries have been exhausted~~, and process from step 5;
4. the Communications Hub shall create an Alert with the Alert Code field set to 0x8F8A as specified in Section 11.7, set `firmwareVersion` to the OTA Upgrade Image File version, include the `transferResponseCode` from the previous step and process from step 6;
5. the Communications Hub shall create an Alert with the Alert Code field set to 0x8F89 as specified in Section 11.7, set `firmwareVersion` to the OTA Upgrade Image File version, include the `transferResponseCode` from the previous step and process from step 6;
6. the Communications Hub shall send the Alert created in the previous step.

The ESME / GSME / HCALCS shall verify the Upgrade Image contained within the OTA Upgrade Image according to Section 11.2.5, and the ESME / GSME shall update its Event Log with the outcome of that verification.

If the verification is successful, the ESME / GSME / HCALCS 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 Distribution Receipt Alert, as detailed in Section 11.2.6.

On receipt of a Firmware Distribution Receipt Alert, the Supplier may verify the cryptographic protection as specified in Section 6.7.3.

Additionally, the Supplier may verify that the Manufacturer Image received by the Device is that intended by comparing the Manufacturer Image Hash in the Firmware Distribution Receipt Alert, with the Hash which it calculates over the Manufacturer Image provided.

Formatted: Font: Italic

Formatted: Font: Italic

Formatted: Font: Italic

Formatted: Font: Italic

[27] ZigBee Document 09-5264r23

11.5 CS06 Activate Firmware

This Use Case covers the activation of a Firmware Image for ESME, GSME, CH and HCALCS.

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
Service Reference	N/A for Communications Hub (this Command does not have a Service Reference for CH) 11.3 for ESME, GSME and HCALCS
Valid Target Device(s)	ESME / GSME / CH / HCALCS
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 / HCALCS 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

Table 11.5: Use Case Cross References for CS06 Activate Firmware

11.5.1 Pre-conditions

None.

11.5.2 Detailed Steps

11.5.2.1 Construction of Command

Activate Firmware Command Payloads shall be constructed according to the requirements of Section 11.5.2.3 and populated as specified in Table 11.5.2.1.

MAC Header, Grouping Header, KRP Signature and ACB-SMD MAC shall be populated as 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			
`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
`originatorCounter`	`INTEGER (0..18446744073709551615)`	The Originator Counter as in the Grouping Header of the Command	Mandatory	
`executionDateTime`	GeneralizedTime	The date-time at which the Command is to be executed, if future dated	OPTIONAL	

Table 11.5.2.1: `@ActivateFirmware.CommandPayload` population

11.5.2.2 Device processing of Command and Response handling

The Device receiving an Activate Firmware Command shall undertake processing steps in the sequence defined in this Section 11.5.2.2.

The Device shall:

1. undertake Command Authenticity and Integrity Verification as required for a Command of the SME.C.C Message Category;
2. if `executionDateTime` is present then the Device shall:
   - record `manufacturerImageHash`, `originatorCounter` and `executionDateTime`;
   - construct and send a Response where `executionOutcome` is not present. Grouping Header is constructed and Response Cryptographic Protection is applied as required for a Response of the SME.C.C Message Categories; and
   - at the date-time specified in `executionDateTime`, or immediately if the `executionDateTime` is in the past, undertake the processing from step 3.

If `executionDateTime` is not present then the Device shall continue processing from step 3 immediately;

3. if the Device does not have a stored Manufacturer Image then set `activateImageResponseCode` to `noImageHeld` and process from step 7;
4. calculate Manufacturer Image Hash. If the calculated value does not equal `manufacturerImageHash` then the Device shall set `activateImageResponseCode` to `hashMismatch` and process from step 7;
5. attempt to activate Manufacturer Image. If the activate fails then the Device shall set `activateImageResponseCode` to `activationFailure` and process from step 7;
6. set `activateImageResponseCode` to `success`;
7. populate the `executionOutcome` according to the requirements of Section 11.5.2.3 using the `activateImageResponseCode` value produced by the processing in this Section 11.5.2.2, the value of `originatorCounter` from the Command and the version of firmware now in operation to populate `firmwareVersion`;
8. construct Grouping Header and apply the Response Cryptographic Protection required for a Response / Alert of the SME.C.C / SME.A.C Message Categories respectively. In such an Alert, the Message Code shall be 0x00CA. The Response / Alert shall be addressed to the Business Originator of the Corresponding Command. If `activateImageResponseCode` is `success` then `alertCode` shall be 0x8F66 else `alertCode` shall be 0x8F67; and
9. send the Response if `executionDateTime` was not present in the Command or send the Alert if `executionDateTime` was present in the Command.

On receipt of the Response, the recipient may undertake the 'Response Recipient Verification' for Responses of type SME.C.C. or for Alerts of type SME.A.C, dependent upon the Message received.

11.5.2.3 Activate Firmware Command, Response and Alert Payloads – structure definition

Each instance of `@ActivateFirmware.CommandPayload` and of `@ActivateFirmware.ResponsePayload` and of `@ActivateFirmware.AlertPayload` shall be an octet string containing the DER encoding of the populated structure defined in this Section 11.5.2.3 which specifies the structure in ASN.1 notation.

```
ActivateFirmware DEFINITIONS ::= BEGIN

CommandPayload ::=                                SEQUENCE
{
   -- specify the hash of the Manufacturer Image to be activated
   manufacturerImageHash                          OCTET STRING,

   -- the Originator Counter as in the Grouping Header of the Command
   originatorCounter                             INTEGER (0..
18446744073709551615),

  -- the date-time at which the Command is to execute, if future dated
  executionDateTime                              GeneralizedTime OPTIONAL
}

ResponsePayload ::=                               CHOICE
{
   -- if the Command is future dated, the Response will not have any details of
   -- execution (those will be in the subsequent alert)
   commandAccepted                                NULL,
```

```
   -- if the Command is for immediate execution, the Response will detail the
   -- outcomes
   executionOutcome                                ExecutionOutcome
}

AlertPayload ::=                                   SEQUENCE
{
   -- specify the Alert Code
   alertCode                                       INTEGER(0..4294967295),

   -- specify the date-time of execution
   executionDateTime                               GeneralizedTime,

   -- the Originator Counter as in the Grouping Header of the corresponding Command
   originatorCounter                               INTEGER (0..
18446744073709551615),

   -- detail what happened when the future dated command was executed
    executionOutcome                               ExecutionOutcome
}

ExecutionOutcome ::=                               SEQUENCE
{
   -- Specify whether the activation was successful or not
   activateImageResponseCode                       ActivateImageResponseCode,

   -- Specify the Device's now current firmware version. The value shall be four
octets in length and shall correspond to the File Version field in the ZSE OTA
Header structure.
   firmwareVersion                                 OCTET STRING
}

ActivateImageResponseCode::= INTEGER
{
   success                                         (0),
   noImageHeld                                     (1),
   hashMismatch                                    (2),
   activationFailure                               (3)
}

END
```

11.6 CS05c Distribute Firmware to PPMID

This Use Case covers the distribution of an OTA Upgrade Image that is intended for a PPMID to the PPMID.

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Command
Message Type Category	None – this is a Variant Message
Capable of future dated invocation?	No
Protection Against Replay Required?	No
Service Reference	11.4
Valid Target Device(s)	PPMID
Valid Business Originator role(s) for Command invocation (and so, for DLMS COSEM Commands, which Application	Supplier

Formatted: Font: Font color: Black

Cross Reference	Value
Association 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	WAN Provider Specific to Communications Hub; ZigBee OTA from Communications Hub to PPMID

Table 11.6: Use Case Cross References for CS05c Distribute Firmware to PPMID

11.6.1 Pre-conditions

None.

11.6.2 Detailed Steps

Italicised terms in this Section 11.6.2 shall have the meaning specified in ZSE.

The PPMID shall use the ZCL *Image Block Request* and *Image Block Response* commands to retrieve available OTA images.

The PPMID 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~~.~~:

1. The Communications Hub shall ~~extract the Manufacturer Image from the Upgrade Image and~~ update its OTA Cluster to reflect the availability of the OTA U~~u~~pgrade I~~i~~mage;
2. the Communications Hub, as OTA Server, shall indicate availability of an OTA U~~u~~pgrade I~~i~~mage for PPMID by sending a ZSE *Image Notify* command;
3. the PPMID should include its *Hardware Version* in the *Query Next Image Request Command*; the Communications Hub shall verify whether the PPMID *Hardware Version* is present in the *Query Next Image Request Command;*
   - if the *Hardware Version* is present and does not fall between~~match~~ the *minimum* and *maximum Hardware Version* in the OTA Header the Communications Hub shall set `transferResponseCode` to ~~`hardware`~~`hardwareVersionMismatch` as specified in Section 11.7 and process from step 7;
   - if the *Hardware Version* is not present, or is present and falls between~~matches~~ the minimum and maximum *Hardware Version* in the OTA Header, the Communications Hub shall make the OTA Upgrade Image available to the PPMID in the *Query Next Image Response Command*;
4. the PPMID shall download an OTA U~~u~~pgrade I~~i~~mage when it is aware of the availability of a suitable OTA U~~u~~pgrade I~~i~~mage using the *QueryNextImage* and *Image Block/Page* commands specified in the *OTA Cluster* specification[28];

Formatted: Font: Not Italic

Formatted: Font: Not Italic

Formatted: Font: Italic

Formatted: Font: Italic

Formatted: Font: Not Italic

Formatted: Font: Not Italic

[28] ZigBee Document 09-5264r23

5. upon the reception of the *Upgrade End Request Command* from the PPMID, the Communications Hub shall:
   - set `transferResponseCode` to `fileTransferSuccess` if the transfer of the OTA Upgrade~~Manufacturer~~ Image has completed successfully and process from step 6;
   - set `transferResponseCode` to `fileTransferFailure` if the transfer of the OTA Upgrade~~Manufacturer~~ Image has failed~~ and all transfer retries have been exhausted~~, and process from step 7;
6. the Communications Hub shall instruct the PPMID to activate the firmware Image as defined in Section 11.2.1 and create an Alert with the Alert Code field set to 0x8F8A as specified in Section 11.7, set `firmwareVersion` to the OTA Upgrade Image File version, include the `transferResponseCode` from the previous step and process from step 8;
7. the Communications Hub shall create an Alert with the Alert Code field set to 0x8F89 as specified in Section 11.7, set `firmwareVersion` to the OTA Upgrade Image File version, include the `transferResponseCode` from the previous step and process from step 8;
8. the Communications Hub shall send the Alert created in the previous step; and
9. upon receipt of the firmware activation instruction from the Communications Hub the PPMID shall activate the firmware and send an Alert with Alert Code 0x8F8B as defined in Section 11.9~~11.8.4~~ to this effect.

Formatted: List Paragraph,List para,Dot pt,No Spacing1,List Paragraph Char Char Char,Indicator Text,Numbered Para 1,List Paragraph1,Bullet 1,Bullet Points,MAIN CONTENT,List Paragraph12,List Para, Numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 0.63 cm + Indent at: 1.27 cm, Tab stops: Not at 1 cm + 9.5 cm

Field Code Changed

11.7 CCS08 Firmware Transfer Alert

11.7.1 Description

This Section 11.710.2.2.4 covers the creation and processing of Alerts resulting from the requirements contained in Section 11.4.2 and Section 11.6.2.

11.7.2 Use Case Cross References

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Alert
Message Type Category	SME.A.C
Capable of future dated invocation?	No
Protection Against Replay Required?	No
Service Reference	N/A
Valid Initiating Device(s)	CHF
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

Table 11.7.2: Use Case Cross References for CCS08 Firmware Transfer Alert

Formatted: Normal, Tab stops: Not at 1 cm + 9.5 cm

11.7.3 Construction of Alert~~s~~

Grouping Header and SMD Signature shall be populated as required for an Alert of the SME.A.C Message Category. The Message Code shall be 0x0128.

The Business Originator ID field in the Grouping Header shall contain the CHF's Entity Identifier.

The Business Target ID field in the Grouping Header shall contain the Entity Identifier in the CHF's `{accessControlBroker, digitalSignature, management}` Trust Anchor Cell.

The Alert Payload shall be constructed according to the requirements of Section 11.7.4 and the Alerts shall be populated as specified in Table 11.7.3.

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
`@FirmwareTransferAlert.AlertPayload`	`SEQUENCE`			
`alertCode`	`INTEGER`	0x8F8A 0x8F89	Mandatory	(1) 0x8F8A: OTA Firmware Image Delivery to the target Devices successful; (2) 0x8F89: OTA Firmware Image Delivery to the target Device failed;
`executionDateTime`	`GeneralizedTime`	The date-time at which this Alert was created	Mandatory	
`otherDeviceEntityIdentifier`	`OCTET STRING`	Entity Identifier	Mandatory	The Entity Identifier of the ESME / GSME / HCALCS / PPMID the firmware upgrade is targeted at
`firmwareVersion`	~~`OCTETT`~~`OCTET STRING`	ZSE firmware version	Mandatory	The firmware version of the transferred or discarded Upgrade Image in ZSE format
`TransferResponseCode`	`INTEGER`	`fileTransferSuccess (0), imageDiscarded(1), ~~hardware~~hardwareVersionMismatch(2), fileTransferFailure (3)`	Mandatory	(a) 0x8F8A: fileTransferSuccess (0) (b) 0x8F89: imageDiscarded (1); ~~hardware~~hardwareVersionMismatch (2); fileTransferFailure (3)

Formatted: Font: Arial

`Table 11.7.3: @FirmwareTransferAlert.AlertPayload population`

11.7.4 Firmware Transfer Alert – structure definition

Each instance of `FirmwareTransferAlert.AlertPayload` shall be an octet string containing the DER encoding of the populated structure defined in this Section 11.7.4 which specifies the structure in ASN.1 notation.

```
FirmwareTransferAlert  DEFINITIONS ::= BEGIN

AlertPayload ::=                                SEQUENCE
{
   -- specify the Alert Code
   alertCode                                    INTEGER(0..4294967295),

   -- specify the date-time of execution
   executionDateTime                            GeneralizedTime,

   -- specify the 64 bit Entity Identifier of the Device the upgrade has been
   -- targeted at
   otherDeviceEntityIdentifier                  OCTET STRING,

   -- specify the upgrade image version if replaced or discarded by the
   -- Communications Hub or if the transfer to the target device succeeded or
   -- failed. The value shall be four octets in length and shall correspond to
   -- the FileVersion field in the ZSE OTA Header structure
   firmwareVersion                              OCTET STRING,

   -- specify the outcome
   transferResponseCode                         TransferResponseCode
}

TransferResponseCode ::=                        INTEGER
{
   fileTransferSuccess                          (0),
   imageDiscarded                               (1),
   ~~hardware~~hardwareVersionMismatch          (2),
   fileTransferFailure                          (3)
}

END
```

11.8 CS08 Read PPMID/HCALCS Firmware Version

11.8.1 Description

This Section 11.8.1 covers the creation and processing of Commands and Responses related to reading the Firmware Version~~, and the sending of Alerts resulting from the requirements contained in Section 11.6.2~~. The Message Code shall be 0x0129.

11.8.2 Use Case Cross References

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Command and~~,~~ Response ~~and Alert~~
Message Type Category	SME.C.NC~~; SME.A.C~~
Capable of future dated invocation?	No
Protection Against Replay Required?	No
Service Reference	11.2
Valid Target Device(s)	PPMID, HCALCS
Valid Business Target role(s) for Alert	Access Control Broker
Valid Business Originator role(s) for Command invocation (and so, for DLMS COSEM Commands, which Application Association is to	Access Control Broker~~, Supplier~~

Cross Reference	Value
be used for delivery of the Command to the Device) [Remote Party Messages Only]	
Valid initiating Device type(s) [HAN Only Messages]	N/A
Protocol	ASN.1

Table 11.8.2: Use Case Cross References for CS08 Read PPMID/HCALCS Firmware Version

Formatted: Normal, Tab stops: Not at 1 cm + 9.5 cm

11.8.3 Construction of Command and Response

11.8.3.1 Construction of Commands

'Read PPMID/HCALCS Firmware Version' Command Payloads shall be constructed as specified in ~~this~~ Section 11.8.4~~11.8.3.1~~ and Cryptographic Protection shall be applied as required for a Command of the relevant Message Category.

11.8.3.2 Device processing of Commands and Response handling

The Device receiving a 'Read PPMID/HCALCS Firmware Version' Command shall undertake processing steps in the sequence defined in this Section 11.8.3.2.

In processing a 'Read PPMID/HCALCS Firmware Version' Command, the Device shall:

1. undertake Command Authenticity and Integrity Verification as required for a Command of this Message Category~~.~~;
2. if the version of firmware in operation is available, set `firmwareVersion` to the version of firmware in operation; set `firmwareReadCode` to `firmwareReadSuccess` and process from Step 4;
3. if the version of firmware in operation is not available, set `firmwareVersion` to 0x00000000; set `firmwareReadCode` to `firmwareReadFailure` and process from Step 4; and
4. create a Response according to the requirements of Section 11.8.4~~11.8.5~~, apply the Response Cryptographic Protection required for a Response of the relevant Message Category, and send the Response.

Field Code Changed

~~11.8.4 Construction of Alerts~~

~~Grouping Header and SMD Signature shall be populated as required for an Alert of the SME.A.C Message Category.~~

~~The Business Originator ID field in the Grouping Header shall contain the PPMID's Entity Identifier.~~

~~The Business Target ID field in the Grouping Header shall contain the Entity Identifier in the PPMID's `{accessControlBroker, digitalSignature, management}` Trust Anchor Cell.~~

~~The Alert Payload shall be constructed according to the requirements of Section 11.8.5 and the Alerts shall be populated as specified in Table 11.8.4.~~

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
@ReadPpmidHcalcsFirmwareVersion.AlertPayload	SEQUENCE			
alertCode	INTEGER	0x8F8B	Mandatory	
executionDateTime	Generalized Time	The date-time at which this Alert was created	Mandatory	
firmwareVersion	OCTETTOCTET STRING	ZSE firmware version	Mandatory	The firmware version in operation in ZSE format or 0x00000000 if reading the firmware version failed
ActivateImageResultCode	INTEGER	activationSuccess (0), activationFailure (1)	Mandatory	

11.8.5 11.8.4 Read PPMID/HCALCS Firmware Version Command and, Response and Alert Payloads – structure definition

Each instance of `@ReadPpmidHcalcsFirmwareVersion.CommandPayload` and, `ReadPpmidHcalcsFirmwareVersion.ResponsePayload` and `ReadPpmidHcalcsFirmwareVersion.AlertPayload` shall be an octet string containing the DER encoding of the populated structure defined in this Section 11.8.4 11.8.5 which specifies the structure in ASN.1 notation.

Formatted: Font: (Default) Arial

Field Code Changed

```
ReadPpmidHcalcsFirmwareVersion                    DEFINITIONS ::= BEGIN

CommandPayload ::=                                NULL

ResponsePayload ::=                               SEQUENCE
{
   -- Specify the Device's now current firmware version. The value shall be four
   -- octets in length and shall correspond to the File Version field in the ZSE
   -- OTA Header structure. Set to 0x00000000 if reading failed.
   firmwareVersion                                OCTET STRING,

   -- specify whether the reading of the firmware was successful.
   firmwareReadCode                               FirmwareReadCode
}

AlertPayload ::=                                  SEQUENCE
{
   -- specify the Alert Code
   alertCode                                      INTEGER(0..4294967295),

   -- specify the date-time of execution
   executionDateTime                              GeneralizedTime,

   -- Specify the Device's now current firmware version. The value shall be four
   -- octets in length and shall correspond to the File Version field in the ZSE
   -- OTA Header structure.
   firmwareVersion                                OCTET STRING,
```

```
   -- Specify whether the activation was successful or not
   activateImageResultCode                          ActivateImageResultCode
}

ActivateImageResultCode::= INTEGER
{
   activationSuccess                                (0),
   activationFailure                                (1)
}

FirmwareReadCode ::=                                INTEGER
{
   firmwareReadSuccess                              (0),
   firmwareReadFailure                              (1)
}

END
```

11.9 CS09 PPMID Firmware Activation Alert

11.9.1 Description

This Section 11.9 covers the construction of the PPMID Firmware Activation Alert. The Message Code shall be 0x012B.

11.9.2 Use Case Cross References

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Alert
Message Type Category	SME.A.C
Capable of future dated invocation?	No
Protection Against Replay Required?	Yes
Service Reference	N/A
Valid Target Device(s)	N/A
Valid Business Target role(s) for Alert	Access Control Broker
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]	N/A
Valid initiating Device type(s) [HAN Only Messages]	N/A
Protocol	ASN.1

Table 11.9.2: Use Case Cross References for CS09 PPMID Firmware Activation Alert~~8 Read PPMID/HCALCS Firmware Version~~

11.9.3 Construction of PPMID Firmware Activation Alert

Formatted: Outline numbered + Level: 3 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 0.25 cm + Indent at: 1.52 cm

Grouping Header and SMD Signature shall be populated as required for an Alert of the SME.A.C Message Category.

The Business Originator ID field in the Grouping Header shall contain the PPMID's Entity Identifier.

The Business Target ID field in the Grouping Header shall contain the Entity Identifier in the PPMID's `{accessControlBroker, digitalSignature, management}` Trust Anchor Cell.

The Alert Payload shall be constructed according to the requirements of Section 11.9.4 and the Alerts shall be populated as specified in Table 11.9.3.

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
`@PpmidFirmwareActivation.AlertPayload`	`SEQUENCE`			
`alertCode`	`INTEGER`	0x8F8B	Mandatory	
`executionDateTime`	`GeneralizedTime`	The date-time at which this Alert was created	Mandatory	
`firmwareVersion`	`OCTET STRING`	ZSE firmware version	Mandatory	The firmware version in operational in ZSE format or 0x00000000 if reading the firmware version failed
`ActivateImageResultCode`	`INTEGER`	`activationSuccess(0)`, `activationFailure(1)`	Mandatory	

Table 11.9.3: `@PpmidFirmwareActivation.AlertPayload` population

Formatted: Font: Arial

11.9.4 PPMID Firmware Activation Alert Payloads – structure definition

Each instance of `@PpmidFirmwareActivation.AlertPayload` shall be an octet string containing the DER encoding of the populated structure defined in this Section 11.9.4 which specifies the structure in ASN.1 notation.

```
PpmidFirmwareActivation                          DEFINITIONS ::= BEGIN


AlertPayload ::=                                 SEQUENCE
{
   -- specify the Alert Code
   alertCode                                     INTEGER(0..4294967295),

   -- specify the date-time of execution
   executionDateTime                             GeneralizedTime,

   -- Specify the Device's now current firmware version. The value shall be four
   -- octets in length and shall correspond to the File Version field in the ZSE
   -- OTA Header structure.
   firmwareVersion                               OCTET STRING,

   -- Specify whether the activation was successful or not
   activateImageResultCode                       ActivateImageResultCode
}

ActivateImageResultCode::= INTEGER
{
```

```
   activationSuccess                                (0),
   activationFailure                                (1)
}

FirmwareReadCode ::=                                INTEGER
{
   firmwareReadSuccess                              (0),
   firmwareReadFailure                              (1)
}

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 5759[29] 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 shall:

- be compliant with IETF RFC 5759 and so with IETF RFC 5280. In adherence with the requirements of IETF RFC 5759, 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 5480[30];
- 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 16 octets in length;
- contain a `subjectKeyIdentifier` which shall be marked as non-critical;
- contain a `certificatePolicies` extension containing at least one `CertPolicyId` which shall be marked as critical. A Device shall reject any Certificate where the value in any `CertPolicyId` is not a valid Object Identifier for a Certificate Policy allowed under the Smart Energy Code for the Device's operating state. For clarity and in adherence with IETF RFC 5280, Certification Path Validation undertaken by Devices shall interpret this extension. Devices shall also interpret this extension when processing a Command with Message Code 0x000B (CS02d Update Device Certificates on Device);
- 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 issuer field whose contents are identical to the Security Credential Document's signer's subject field in the signer's Security Credential Documents; and

[29] http://tools.ietf.org/html/rfc5759
[30] http://tools.ietf.org/html/rfc5480

- 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.

12.2 Requirements applicable to Organisation Certificates only

All Organisation Certificates that are Authorised for use by Devices shall:

- have a fixed expiration date in the `notAfter` field which shall not be `GeneralizedTime` value of 99991231235959Z;
- contain a non-empty subject field which shall contain an `X520OrganizationalUnitName` whose value shall be set to the `RemotePartyRole` that this Certificate allows the subject of the Certificate to perform; and
- contain a single Public Key except where the `RemotePartyRole = root`. Where the `RemotePartyRole = root`, the Certificate shall contain two public keys. The second public key shall be referred to as the Contingency Key[31] and shall be present in the `WrappedApexContingencyKey` extension with the meaning of IETF RFC 5934[32]. The Contingency Key shall be Encrypted as per the requirements of Section 13.3.5.8.1.

12.3 Requirements applicable to Certificates where `RemotePartyRole = root` or `issuingAuthority`

All Remote Parties' Certificates that:

- are Authorised within the APKI for use by Devices; and
- have a `X520OrganizationalUnitName` whose value is either `root` or `issuingAuthority`

shall:

- have a `keyUsage` with a value of `keyCertSign` and `cRLSign`. For clarity, Devices are not required to use the associated Public Keys in relation to the `cRLSign` `keyUsage` and shall disregard a `keyUsage` of `cRLSign` in any processing;
- where `X520OrganizationalUnitName = issuingAuthority`:
  - contain at least one `CertPolicyId` in the `certificatePolicies` extension that refers to the OID(s) valid for usage in GB Smart Metering;
  - contain the `basicConstraints` extension, with values `cA` = True, and `pathLen` = 0. This extension shall be marked as critical;
- where `X520OrganizationalUnitName` = `root`:
  - contain a single `CertPolicyId` in the `certificatePolicies` extension that refers to the OID for `anyPolicy`; and

[31] The Contingency Key is a second public key held in the Root Certificate (and protected with an encryption key). Its purpose is to allow the validation of a specific command that allows direct replacement of the Root Trust Anchor or the direct replacement of any other Trust Anchor cell. 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.

[32] Housley, R., Ashmore, S., and C. Wallace, 'Trust Anchor Management Protocol (TAMP)', RFC 5934, August 2010. https://tools.ietf.org/html/rfc5934

- contain the basicConstraints extension, with the value cA = True and pathLen absent (unlimited). This extension shall be marked as critical.

Note that Certificates where RemotePartyRole is root do not have to contain a Subject Unique ID. Thus, the Device will not have a value provided to it for this field in the {root, keyCertSign, management} Trust Anchor Cell. For consistency, Devices shall record a value of 0x0000000000000000 for Subject Unique ID in the {root, keyCertSign, management} Trust Anchor Cell, and shall use this in responding to Security Credentials Commands, where values for Subject Unique ID are required from this Trust Anchor Cell.

12.4 Requirements applicable to Certificates where RemotePartyRole is neither root nor issuingAuthority

All Remote Parties' Certificates that:

- are Authorised within the APKI for use by Devices; and
- have a X520OrganizationalUnitName whose value is not root and is not issuingAuthority

shall:

- contain a Subject Unique ID whose value shall be the 8 octet Entity Identifier of the subject of the Certificate;
- have a keyUsage with a value of only one of digitalSignature or keyAgreement; and
- contain a single policyIdentifier in the certificatePolicies extension that refers to the OID applicable to the environment the Certificate has been issued in.

12.5 Requirements applicable to Device Certificates

All Device Certificates that are Authorised within the APKI for use by Devices shall:

- not have a well-defined expiration date and so the notAfter field shall be assigned the GeneralizedTime value of 99991231235959Z;
- have an empty SubjectName;
- have a keyUsage with a value of only one of digitalSignature or keyAgreement;
- contain a single policyIdentifier in the certificatePolicies extension that refers to the OID applicable to the environment the Device Certificate has been issued in;
- contain a SubjectAltName extension which shall contain a single GeneralName of type OtherName that is further sub-typed as a HardwareModuleName (id-on-HardwareModuleName) as defined in IETF RFC 4108[33]. The hwSerialNum field shall be set to the Device's Entity Identifier. In adherence to IETF RFC 5280, SubjectAltName shall be marked as critical; and
- contain a single Public Key.

[33] http://tools.ietf.org/html/rfc4108

12.6 Device processing of Certificates

In relation to Certificates, Devices shall:

- accept unexpected (not required by the GBCS) certificate extensions and shall ignore silently non-critical unrecognized certificate extensions;
- in adherence with the requirements of IETF RFC 5280, reject any certificate containing unrecognized critical certificate extensions; and
- reject any certificate containing either policy mappings or name constraints.

13 Managing Security Credentials on Devices

Requirements in:

- Section 13.3 and in Use Case CS02b Update Security Credentials Command, Response and Alert;
- Section 13.4 and in Use Case CS02c Issue Security Credentials;
- Section 13.5 and in Use Case CS02d Update Device Certificates on Device; and
- Section 13.6 and in Use Case CS02e Provide Device Certificates from Device;

shall apply to an SAPC as if it were an ESME.

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:

- Section 13.2 - CS02a Provide Security Credential Details Command and Response;
- Section 13.3 - CS02b Update Security Credentials Command, Response and Alert;
- Section 13.4 - CS02c Issue Security Credentials;
- Section 13.5 - CS02d Update Device Certificates on Device;
- Section 13.6 - CS02e Provide Device Certificates from Device;
- Section 13.7- Pair-wise Authorisation of Devices (covered by various Join / Unjoin Use Cases);
- Section 13.8 - GPF Device Log Backup and Restore (GCS59 and GCS62); and
- Section 13.9 - CS02f Provide Security Credential Details Command and Response; and
- Section 13.10 - CS02g Update Load Controller Security Credentials Command, Response and Alert.

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 Certification Requests (and therefore generate new Public-Private Key Pairs) is covered in Section 13.4;
- the Command to Devices for the Device to replace a current Device Certificate with a new Device Certificate resulting from a Device Certification Request is covered in Section 13.5, as are the related requirements for the capability to store such Certificates; 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 Protocol (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.

13.2 CS02a Provide Security Credential Details Command and Response

13.2.1 Description

This Section covers the creation, validation and processing of (i) CS02a Provide Security Credential Details Commands and (ii) Responses to such Commands.

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
Service Reference	6.24
Valid Target Device(s)	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

Table 13.2.2: Use Case Cross References for Provide Security Credential Details Command and Response

13.2.3 Common Requirements

13.2.3.1 Summary – informative

Remote Party Security Credentials are provided to Devices as Certificates which are X.509 based, DER encoded ASN.1 structures. Hence, the Command's structure is specified using ASN.1 with DER encoding to be applied to Command instances. Note that the details provided in the Response include the related Execution Counter details held on the Device.

13.2.3.2 The 'CS02a Provide Security Credential Details' Command and Response

This Section 13.2.3.2 summarises the structure of the CS02a Provide Security Credential Details Command.

If protected by an Access Control Broker MAC as per Section 13.2.4.2, a CS02a Provide Security Credential Details Command shall be the concatenation:

MAC Header || Grouping Header || `@ProvideSecurityCredentialDetails.Command` || 0x00 || ACB-SMD MAC

If protected by a KRP Signature as per Section 13.2.4.2, a CS02a Provide Security Credential Details Command shall be the concatenation:

Grouping Header || `@ProvideSecurityCredentialDetails.Command` || 0x40 || KRP Signature

If an SMD Signature is required as per Section 13.2.4.5, a CS02a Provide Security Credential Details Response shall be the concatenation:

Grouping Header || `@ProvideSecurityCredentialDetails.Response` || 0x40 || SMD Signature

If an SMD Signature is not required as per Section 13.2.4.5, a CS02a Provide Security Credential Details Response shall be the concatenation:

MAC Header || Grouping Header || `@ProvideSecurityCredentialDetails.Response` || 0x00 || SMD-KRP MAC

Where:

- `@ProvideSecurityCredentialDetails.Command` and `Response` shall each be an octet string containing the DER encoding of the populated ASN.1 structure (as laid out in Section 13.2.3.3);
- 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 SMD or KRP Signature is not present;
- KRP Signature and ACB-SMD MAC are as defined in Section 13.2.4.2;
- SMD Signature and SMD-KRP MAC are as defined in Section 13.2.4.5; and
- MAC Header and Grouping Header are as defined in Section 7.2.

13.2.3.3 The @ProvideSecurityCredentialDetails.Command and @ProvideSecurityCredentialDetails.Response structure definition

Each instance of `@ProvideSecurityCredentialDetails.Command` and of `@ProvideSecurityCredentialDetails.Response` shall be an octet string containing the DER[34] encoding of the populated structure defined in this Section 13.2.3.3 which specifies the structure in ASN.1 notation[35].

```
ProvideSecurityCredentialDetails DEFINITIONS ::= BEGIN

Command ::=                                 SEQUENCE
{
-- Identify which of the Public Keys on the Device is to be used in verifying the Signature or MAC
-- (so defining the nature of the verification by way of the KeyUsage parameter held on the
-- Device for the Public Key so identified).

authorisingRemotePartyTACellIdentifier        TrustAnchorCellIdentifier,

-- List the Remote Party Role(s) for which credential details are required

remotePartyRolesCredentialsRequired           SEQUENCE OF RemotePartyRole
}

Response ::=                                  SEQUENCE OF RemotePartyDetails

RemotePartyDetails ::=                        SEQUENCE
{

-- Which Remote Party do these details relate to?
remotePartyRole                               RemotePartyRole,

-- statusCode shall be success unless the role is not valid on this type of Device or there is a processing failure
statusCode                                    StatusCode,


-- What is the current Update Security Credentials Execution Counter on the Device for this role, where there is such a number
for this role (see Table 13.2.4.4)?

currentSeqNumber                               SeqNumber OPTIONAL,

-- What are the details held on the Device for each of the Cells related to this role? The list shall have between one and
```

[34] https://www.itu.int/rec/T-REC-X.690/en
[35] https://www.itu.int/rec/T-REC-X.680/en

```
-- three entries (e.g. there will be one if role is transitional change of supplier; there may be three if role is supplier)

trustAnchorCellsDetails                       SEQUENCE OF TrustAnchorCellContents OPTIONAL
}

SeqNumber ::=                                 INTEGER (0.. 18446744073709551615)

TrustAnchorCellContents ::=                   SEQUENCE
{
-- To what cryptographic use can the Public Key in this Cell be put? Some Remote Party Roles
-- (e.g. supplier) can have more than one Public Key on a Device and each one would only have
-- a single cryptographic use.

trustAnchorCellKeyUsage                       KeyUsage,

-- trustAnchorCellUsage is to allow for multiple Public Keys of the same keyUsage for the same Remote
-- Party Role. This will be absent except where used to refer to the Supplier Key Agreement Key.
-- This Key is used solely in relation to validating Supplier generated MACs on Prepayment Top Up transactions.

trustAnchorCellUsage                          CellUsage DEFAULT management,

-- The existingSubjectUniqueID shall be the 64 bit Entity Identifier of the Security Credentials in this Trust Anchor Cell.

existingSubjectUniqueID                       OCTET STRING,

-- The APKI requirements mean that KeyIdentifier attributes will all be 8 byte SHA-1 Hashes.
-- existingSubjectKeyIdentifier shall be set accordingly based on the contents of the Trust Anchor Cell

existingSubjectKeyIdentifier                  OCTET STRING
}

TrustAnchorCellIdentifier ::=                 SEQUENCE
{
-- Which Remote Party Role does this Cell relate to?

trustAnchorCellRemotePartyRole                RemotePartyRole,

-- To what cryptographic use can the Public Key in this Cell be put? Some Remote Party Roles
-- (e.g. supplier) can have more than one Public Key on a Device and each one would only have
-- a single cryptographic use.

trustAnchorCellKeyUsage                       KeyUsage,

-- trustAnchorCellUsage is to allow for multiple Public Keys of the same keyUsage for the same Remote
```

```
-- Party Role. This may be absent except where use to refer to the Supplier Key
-- Agreement Key used solely in relation to validating Supplier generated MACs on Prepayment Top Up transactions

trustAnchorCellUsage                          CellUsage DEFAULT management
}

CellUsage ::=                                 INTEGER {management(0), prePaymentTopUp(1)}

RemotePartyRole ::=                           INTEGER
{
-- Define the set of Remote Party Roles in relation to which a Device may need to undertake
-- processing. Note that most Devices will only support processing in relation to a subset of these.

root                                          (0),
recovery                                      (1),
supplier                                      (2),
networkOperator                               (3),
accessControlBroker                           (4),
transitionalCoS                               (5),
wanProvider                                   (6),
issuingAuthority                              (7),    -- Devices will receive such Certificates but they do not
                                                      -- need to store them over an extended period



-- The 'other' RemotePartyRole is for a party whose role does not allow it to invoke any Device function apart from
-- UpdateSecurityCredentials. This is to allow for Device functionality to be locked out of usage until a valid
-- Remote Party can be identified e.g. where roles cannot be fixed until a Device is bought in to operation
other                                         (127)

}

-- KeyUsage is only repeated here for ease of reference. It is defined in RFC 5912

KeyUsage ::=                                  BIT STRING
{
-- Define valid uses of Public Keys.

digitalSignature                              (0),
contentCommitment                             (1),    -- not valid for GBCS compliant transactions
keyEncipherment                               (2),    -- not valid for GBCS compliant transactions
dataEncipherment                              (3),
keyAgreement                                  (4),
keyCertSign                                   (5),
```

```
cRLSign                                     (6),
encipherOnly                                (7),
decipherOnly                                (8)     -- not valid for GBCS compliant transactions
}

-- The GBCS only allows for a constrained set of Trust Anchor Cell operations and so the list of possible outcomes
-- is more limited than in IETF RFC 5934. The list below is that more constrained subset

StatusCode ::=                              ENUMERATED {

success                                     (0),

-- trustAnchorNotFound indicates that details of a trust anchor were requested, but the referenced trust anchor
-- is not represented on the Device

trustAnchorNotFound                         (25),

other                                       (127)}


END
```

13.2.4 CS02a 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 CS02a 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 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 since the prePaymentTopUp key pair cannot be used in relation to this command
remotePartyRolesCredentialsRequired	SEQUENCE OF			
RemotePartyRole	INTEGER	root (0) , recovery (1) , supplier (2) ,	Mandatory to have at least one	List the Remote Party Role(s) for which credential details are required

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
		`networkOperator (3)`, `accessControlBroker (4)`, `transitionalCoS (5)`, `wanProvider (6)`		

Table 13.2.4.1: Attribute values for CS02a Provide Security Credentials Command

13.2.4.2 Command Cryptographic Protection

If the Access Control Broker is undertaking this step to apply a MAC, then the Access Control Broker shall undertake the steps in Section 13.2.4.2.1 otherwise:

- the Remote Party originating the Command shall generate a Signature for the Command and set KRP Signature accordingly;
- the Signature, for incorporation in the Command, shall only be generated once all fields of the Grouping Header || `@ProvideSecurityCredentialDetails.Command` are populated as per the requirements for the Command Construction stage; and
- the Remote Party shall use its Private Digital Signing Key to generate the Signature.

13.2.4.2.1 Access Control Broker MAC

If the Access Control Broker is undertaking this step to apply a MAC, then the Access Control Broker shall calculate a MAC using the 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:	0x110000000000 || Grouping Header || `@ProvideSecurityCredentialDetails.Command` || 0x00	

Table 13.2.4.2.1: Calculation of Access Control Broker MAC for CS02a Provide Security Credentials command

13.2.4.3 Command Authenticity and Integrity Verification

The Device shall undertake processing according to the requirements of this Section before undertaking any other processing of the Command.

The checks should be carried out in the order specified. The Device shall cease checking at the point that any one check fails.

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

Table 13.2.4.3: Provide Security Credentials Command authenticity and integrity verification

Should any of the checks detailed in this Section 13.2.4.3 fail then the Device shall:

- generate an entry in the Security Log recording failed Authentication;
- discard the Command without execution and without sending a Response; and
- send an Alert notifying the failed Authentication, constructed as specified in Section 6.2.4.2, populated with the relevant Alert Code from Section 16 , to the Known Remote Party identified by the Security Credentials it holds in the `{supplier, management, digitalSignature}` Trust Anchor Cell.

Where all of the checks detailed in this Section 13.2.4.3 succeed the Device shall process the Command and produce a Response.

13.2.4.3.1 Command Authenticity and Integrity Verification

The Device shall undertake the following checks until either all are successful or one has failed.

1. If `trustAnchorCellUsage` is present this test shall fail.
2. If `trustAnchorCellKeyUsage` = `keyAgreement` then
   ((`trustAnchorCellRemotePartyRole` = `accessControlBroker`) and (the MAC calculated by the Device according to Table 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 identified by `authorisingRemotePartyTACellIdentifier` to verify that KRP Signature is the Digital Signature across Grouping Header || `@ProvideSecurityCredentialDetails.Command`)

   else
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:	0x110000000000 || Grouping Header || `@ProvideSecurityCredentialDetails.Command` || 0x00	

Table 13.2.4.3.1: Calculation of MAC for Provide Security Credential Details Command

13.2.4.4 Response Construction

The Device shall populate Grouping Header according to the requirements of Section 7.2.6.

The `@ProvideSecurityCredentialDetails.Response` shall have the structure defined in Section 13.2.3.3, and the Device shall 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) ,`
		`supplier (2) ,`
		`networkOperator (3) ,`
		`accessControlBroker (4) ,`
		`transitionalCoS (5) ,`
		`wanProvider (6) ,`
			Mandatory if `SEQUENCE` is present	The role to which the credentials in this `SEQUENCE` relate

`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 Execution Counter held by the Device for this role's use of the Update Security Credentials Command. Where this role is `root`, the value of the `anyByContingency` Execution Counter shall be returned. Where this role is `transitionalCoS`, the value of the `transCoSByTransCos` Execution Counter shall be returned. Where this role is `supplier`, the value of the `supplierBySupplier` Execution Counter shall be returned.

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
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 (0)	Only needs to be present for the {supplier, keyAgreement, 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	

Table 13.2.4.4: Attribute values for Provide Security Credentials Response

13.2.4.5 Response Cryptographic Protection

If the Command that triggered this Response was protected by a MAC then the Device shall calculate a MAC using the parameters in Table 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:	0x110000000000 || Grouping Header || `@ProvideSecurityCredentialDetails.Response` || 0x00	

Table 13.2.4.5: Calculation of MAC for CS02a Provide Security Credential Details Response

Otherwise:

- the Device creating the Response shall generate a Signature for the Response and set SMD Signature to the value calculated;
- the Signature, for incorporation in the Response, shall only be generated once all fields of the Grouping Header || Length || `@ProvideSecurityCredentialDetails.Response` are populated, as per requirements for the Response Construction stage; and
- the Device shall use its Private Digital Signing Key to generate the Signature.

13.2.4.6 Response Recipient Cryptographic Verification

If the Response contains a MAC, the Access Control Broker can verify that MAC by calculating a MAC according to the parameters in Table 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:	0x110000000000 || Grouping Header || `@ProvideSecurityCredentialDetails.Response` || 0x00	

Table 13.2.4.6: Calculation of MAC for CS02a Provide Security Credential Details Response Verification

13.3 CS02b Update Security Credentials Command, Response and Alert

13.3.1 Description

This Section 13.3 and Section 13.10 cover the creation, validation and processing of:

- Update Security Credentials Commands;
- Responses to such Commands; and
- Alerts resulting from the future dated execution of such Commands.

The Update Security Credentials Commands shall be:

- used solely to replace Remote Party Security Credentials held in Trust Anchor Cells on Devices;
- supported by any Device that can process Remote Party Messages; and
- 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.

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
Service 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	Supplier Network Operator Access Control Broker

Cross Reference	Value
Association is to be used for delivery of the Command to the Device) [Remote Party Messages Only]	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

Table 13.3.2: Use Case Cross References for Update Security Credentials Command

13.3.3 Command, Response and Alert Structure

13.3.3.1 The Update Security Credentials Command

This Section 13.3.3.1 summarises the structure of the Update Security Credentials Command, which depends on `credentialsReplacementMode` and the `deviceType` of the Device.

If `credentialsReplacementMode` = `anyByContingency` or `anyExceptAbnormalRootByRecovery` then an Update Security Credential Details Command shall be the concatenation:

Grouping Header || `@UpdateSecurityCredentials.CommandPayload` || 0x40 || KRP Signature

If `credentialsReplacementMode` = `accessControlBrokerByACB` and `deviceType` is not `communicationsHubCommunicationsHubFunction` then an Update Security Credentials Command shall be the concatenation:

MAC Header || Grouping Header || `@UpdateSecurityCredentials.CommandPayload` || 0x00 || ACB-SMD MAC

In all other cases, the Update Security Credentials Command shall be the concatenation:

MAC Header || Grouping Header || `@UpdateSecurityCredentials.CommandPayload` || 0x40 || KRP Signature|| ACB-SMD MAC

In these Command structures:

- `@UpdateSecurityCredentials.CommandPayload` shall be an octet string containing the DER encoding of the populated ASN.1 structure (as laid out in Section 13.3.5.11);

- Grouping Header shall be constructed as specified in Section 7.2.7 with Business Originator ID being the Entity Identifier of the Known Remote Party which generated KRP Signature, and with Business Originator Counter being that of the same Known Remote Party;
- KRP Signature shall be generated as specified in Section 6.3.3;
- ACB Grouping Header shall be constructed as specified in Section 7.2.7 with Business Originator ID being the Entity Identifier of the Access Control Broker and Business Originator Counter being that of the Access Control Broker;
- MAC Header shall be constructed as specified in Section 7.2.5; and
- ACB-SMD MAC shall be calculated as specified in Section 6.2.3.

13.3.3.2 The Update Security Credentials Response

An Update Security Credentials Response shall be the concatenation:

Grouping Header || `@UpdateSecurityCredentials.ResponsePayload` || 0x40 || SMD Signature

where:

- `@UpdateSecurityCredentials.ResponsePayload` shall be an octet string containing the DER encoding of the populated ASN.1 structure (as laid out in Section 13.3.5.11);
- Grouping Header in the Response shall be constructed as specified in Section 7.2.7 with Business Target ID being the Entity Identifier specified in the corresponding Command's Grouping Header; and
- SMD Signature shall be generated as specified in Section 6.3.5.

13.3.3.3 The Update Security Credentials Alert

An Update Security Credentials Alert shall be the concatenation:

Grouping Header || `@UpdateSecurityCredentials.AlertPayload` || 0x40 || SMD Signature

where:

- `@UpdateSecurityCredentials.AlertPayload` shall be an octet string containing the DER encoding of the populated ASN.1 structure (as laid out in Section 13.3.5.11);
- Grouping Header in the Alert shall be constructed as specified in Section 7.2.7 with Business Target ID being the Entity Identifier specified in the corresponding Command's Grouping Header;
- the Message Code being 0x00CB; and

- SMD Signature shall be generated as specified in Section 6.3.5.

13.3.3.4 The Update Security Credentials Command, Response and Alert – informative

The `@UpdateSecurityCredentials.CommandPayload` structure has four parts:

- `authorisingRemotePartyControl`: which includes details of what kind of credential replacement this Command is, which Remote Parties are authorising it and information to support Protection Against Replay protections;
- `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 is to be stored in on the Device;
- `certificationPathCertificates`: which is a list of Certification Authority Certificates the Device will need to use in checking that the replacement Certificates were properly issued; and
- `executionDateTime`: which, if present, specifies the date-time at which the `certificates` in the `CommandPayload` are to be used to replace the credentials currently in use on the Device. If this field is not present, the Command shall be executed immediately. If this field has the value equivalent to 'never' (which is '99991231235959Z') the certificate replacement will never happen. This is to allow cancellation of future dated Commands. Note that future dating is not supported where certificates are being replaced in exception conditions.

The `@UpdateSecurityCredentials.Response` structure contains, for immediate execution commands, a list detailing the success of failure of each of the replacements, including details of the parties affected. For future dated commands, `@UpdateSecurityCredentials.AlertPayload` structure contains the list detailing the success, or failure, of each of the replacements, including details of the parties affected.

Section 13.3.5.11 contains narrative for each of the parts of these ASN.1 structures.

Section 18.3.1 provides an illustrative instantiation of `@UpdateSecurityCredentials.CommandPayload` and its corresponding DER encoding.

13.3.4 Updating Security Credentials on a Device – Processing Steps

This Section lays out the requirements for the construction, protection and Authentication of the Update Security Credentials Command Payload, the processing required on the Device of the Command, the construction of the corresponding Response Payload and, where required, the Alert Payload.

13.3.4.1 Command Payload construction

The `@UpdateSecurityCredentials.CommandPayload` shall have the structure defined in Section 13.3.5.11, and the Remote Party 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
`@ UpdateSecurityCredentials.Command ::=`	`SEQUENCE`			
`authorisingRemotePartyControl`	`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 and allow counters / counter caches on the Device to be reset, if the command changes the Remote Party in control

`credentialsReplacementMode`	`INTEGER`	`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 *kindOfCertificate(s)BeingReplacedBypartydoingthereplacement* . For example, `supplierBySupplier` is where a new supplier 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 `credentialsReplacementMod`

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
				e = anyByContingency). The contents provide the symmetric key to decrypt the Contingency Public Key in the (root, digitalSignature, management) Trust Anchor Cell
applyTimeBasedCPVChecks	[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. supplier credentials on the Device have expired without replacement for unforeseen reasons)
authorisingRemotePartyTACellIdentifier	[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. 'authorisingRemotePartyTACellIdentifier' can only be omitted when the Access Control Broker is changing its own Key Agreement credentials
trustAnchorCellRemotePartyRole	INTEGER	root (0), recovery (1) , supplier (2) ,	Mandatory if authorisingRemotePartyTACellIdentifier	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 credentialsReplacementMod

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
		networkOperator (3) , accessControlBroker (4) , transitionalCoS (5) , wanProvider (6)	present	e = anyByContingency and plaintextSymmetricKey is populated with the symmetric key required to decrypt that public key
trustAnchorCellKeyUsage	BIT STRING	digitalSignature (0) if credentialsReplacementMode <> anyByContingency, keyCertSign (5) if credentialsReplacementMode = anyByContingency	Mandatory if authorisingRemotePartyTACellIdentifier present	KRP Signature is a digital signature
trustAnchorCellUsage	INTEGER	management(0)	DEFAULT management	Must be absent 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	OPTIONAL	If the Command is to effect a change of control, then newRemotePartyFloorSeqNumber should be included and will

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
		the update is successful		be the value used to prevent replay of Update Security Credentials Commands, and other Commands, for the new controlling Remote Party
newRemotePartySpecialistFloorSeqNumber	[5] IMPLICIT SEQUENCE OF		OPTIONAL	Some Commands on the Device may use a different Originator 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		OPTIONAL	In some cases, one party acting in one Remote Party Role may be

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			replacing certificates for a different Remote Party Role (e.g. transitionalCoS changing Supplier Credentials). In such cases, Execution Counters need also to be reset for that other 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 did not 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
newRemotePartySpecialistFloorSeqNumber	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 newRemotePartySpecialist	Specify the usage of the 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
			FloorSeqNumber present	
seqNumber	INTEGER	Relevant Originator Counter	OPTIONAL	Specify the associated SeqNumber
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) ,	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
		transitionalCoS (5) , wanProvider (6)		
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: • the deviceType is eSME or gSME; and • the supplier operating the Device wishes to use prepayment top up functionality on the Device, and this is a replacement of the corresponding certificate. Note the certificate specified for use in the {supplier, keyAgreement, prePaymentTopUp} Trust Anchor Cell may be the same key as that specified for the {supplier, keyAgreement, management} Trust Anchor Cell or may be different.
certificationPathCertificates	SEQUENCE OF	The list of certificates needed for Certification Path Validation	At least one Certificate must be present	Provide the certificates needed to undertake Certification Path Validation against the 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
	Certificate			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
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

Table 13.3.4.1: Attribute values for Update Security Credentials Command

13.3.4.2 Command Authenticity and Integrity Verification

The Device shall undertake processing according to the requirements of Section 13.3.5.1.

Should any of the checks detailed in Section 13.3.5.1 fail then the Device shall:

- generate an entry in the Security Log recording failed Authentication;
- discard the Command without execution and without sending a Response; and
- 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:
  - the Trust Anchor Cell {supplier, digitalSignature, management} if the Device's deviceType is not communicationsHubCommunicationsHubFunction; or
  - the Trust Anchor Cell {wanProvider, digitalSignature, management} if the Device's deviceType is communicationsHubCommunicationsHubFunction.

Where all of the checks detailed in Section 13.3.5.1 succeed the Device shall process the Command and produce a Response.

13.3.4.3 Command Processing

Before undertaking any further processing, the Device shall update Execution Counter to the value of `authorisingRemotePartySeqNumber`.

If `executionDateTime` is present then the Device shall:

- record against the `remotePartyRole` (as specified in `authorisingRemotePartyControl`), `authorisingRemotePartyControl`, `replacements`; and `executionDateTime`;
- construct a Response where `executionOutcome` is not present according to the requirements of Section 13.3.4.4; and
- at the date-time specified in `executionDateTime`, or immediately if the `executionDateTime` is in the past undertake the processing of Section 13.3.4.3.1 then construct an Alert according to the requirements of Section 13.3.4.5.

If `executionDateTime` is not present then the Device shall:

- undertake the processing of Section 13.3.4.3.1; and
- construct a Response where `executionOutcome` is present according to the requirements of Section 13.3.4.4.

13.3.4.3.1 `replacements` Processing

For each of the `targetTrustAnchorCell` in `replacements`, the Device shall:

- record the `entityIdentifier` and `subjectKeyIdentifier` currently held in that `targetTrustAnchorCell`;
- attempt to replace the contents of that `targetTrustAnchorCell` using the corresponding `certificate` in `TrustAnchorReplacement`; and
- if the contents of the `replacement` are successfully applied, undertake the processing required by Section 13.3.5.10 in relation to the `RemotePartyRole` for that `targetTrustAnchorCell`.

For clarity, before the contents of a certificate are used to replace the contents of a targetTrustAnchorCell, the Device shall have successfully undertaken the checks in Section 13.3.5.9 in relation to that certificate. Should a check in 13.3.5.9 not succeed then, in line with Section 4.3.2.8.2, the Device shall undertake no further processing of the Command, except for the issuance of a Message containing an `executionOutcome`.

13.3.4.4 Response Construction

The `@UpdateSecurityCredentials.ResponsePayload` shall have the structure defined in Section 13.3.5.11, and the Device shall populate the `executionOutcome`, where present with values according to Table 13.3.4.6.

13.3.4.5 Alert Construction

The `@UpdateSecurityCredentials.AlertPayload` shall have the structure defined in Section 13.3.4, and the Device shall populate the `executionOutcome`, with values according to Table 13.3.4.6.

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`	`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		Mandatory containing zero, one or many occurrences of the following structure	The resulting changes to any Execution Counter 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 Execution Counter values changed on the Device
otherRemotePartyFloorSeqNumber	INTEGER	The corresponding Execution 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

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
seqNumber	INTEGER		Mandatory if newRemotePartySpecialistFloorSeqNumber present	Specify the associated SeqNumber
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

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
trustAnchorCellUsage	INTEGER	{management(0) , prePaymentTopUp(1) }	DEFAULT management	Absent unless: • the deviceType is eSME or gSME; and • the supplier operating the Device wishes to use prepayment top up functionality on the Device, and this is a replacement of the corresponding certificate. Note the certificate specified for use in the {supplier, keyAgreement, prePaymentTopUp} Trust Anchor Cell may be the same key as that specified for the {supplier, keyAgreement, management} Trust Anchor Cell or may be different.
statusCode	ENUMERATED	success (0) , badCertificate (5) , noTrustAnchor (10) , insufficientMemory (17) , 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

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
`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
`replacingSubjectKeyIdentifier`	`OCTET STRING`		Mandatory if `SEQUENCE` is present	For the public key which was to be placed in this Cell

Table 13.3.4.6: Attribute values for `executionOutcome`

13.3.5 Common Requirements

13.3.5.1 Update Security Credentials Command Verification

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 may be carried out in any order. 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

Check Number	Criteria that must be tested by the Device	How the Device shall test the Criteria
2.1	The `targetTrustAnchorCells` all exist on a Device of this type	As specified in Section 13.3.5.4
2.1	The `trustAnchorCellUsage` in the `authorisingRemotePartyTACellIdentifier` must be `ABSENT`	As specified in Section 13.3.4.1
2.2	The `credentialsReplacementMode` is one that can be Authorised by the Remote Party / Parties authorising the Command	As specified in Section 13.3.5.5
2.2	The `replacements` specified are all allowed in this `credentialsReplacementMode`	As specified in Section 13.3.5.6
2.2	The `TrustAnchorCellUsage` in the `targetTrustAnchorCell` in each entry in `replacements` must be `ABSENT` unless it has the value `prePaymentTopUp` as specified in the Notes column of Table 13.3.4.1	As specified in Section 13.3.4.1
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

Table 13.3.5.1: Update Security Credentials Command authenticity and integrity verification

13.3.5.2 Message Code Validation

`CredentialsReplacementMode`	Message Code
`supplierBySupplier`	0x0102
`networkOperatorByNetworkOperator`	0x0103
`accessControlBrokerByACB`	0x0104
`wanProviderByWanProvider`	0x0105
`transCoSByTransCoS`	0x0106
`supplierByTransCoS`	0x0107
`anyExceptAbnormalRootByRecovery`	0x0108
`anyByContingency`	0x0109

Table 13.3.5.2: Message Code validation against `CredentialsReplacementMode`

13.3.5.3 Preventing Replay of Commands

The Protection Against Replay mechanisms for the Update Security Credentials Command shall be that specified in this Section 13.3.5.3 (which is different than that for other GBCS Commands).

For each of `RemotePartyRole` from which the Device can receive a valid Updated Security Credentials Command, the Device shall allocate storage for an Execution Counter which shall be capable of storing a 64 bit unsigned integer and which shall initially be set to the value zero at manufacture. For `transitionalCoS`, the Device may allocate storage for an additional Execution Counter, `credentialsReplacementModes`.

Before executing any Update Security Credentials Command, a Device shall confirm that, if `CredentialsReplacementMode` <> `accessControlBrokerByACB`, then

- (`authorisingRemotePartyTACellIdentifier` is populated in the Command) and (the `authorisingRemotePartySeqNumber` is strictly numerically greater than the Execution Counter the Device has recorded for the `RemotePartyRole` identified in `authorisingRemotePartyTACellIdentifier` and, where relevant for `transitionalCoS`, the `credentialsReplacementMode`);

else

- (the `authorisingRemotePartySeqNumber` is strictly numerically greater than the Execution Counter the Device has recorded for the `accessControlBroker`).

13.3.5.4 Required Trust Anchor Cells by Device Type

The Trust Anchor Cells specified in Section 4.3.2.5 are those required on each Device type and so are the only valid `targetTrustAnchorCells`.

The Device shall ensure that all `targetTrustAnchorCells` specified in the Command instance are valid for the type of Device it is, according to the requirements of Section 4.3.2.5. A Command containing any invalid `targetTrustAnchorCells` shall not be processed any further by the Device.

13.3.5.5 Valid `credentialsReplacementMode` by Remote Party Roles authorising the Command

A Command containing a certain `credentialsReplacementMode` is only Authorised using certain types of Public-Private Key Pairs in certain ways. The Command identifies the Public Key corresponding to the Private Key used by the authorising Remote Party in the `authorisingRemotePartyTACellIdentifier` structure. Table 13.3.5.5 lists the only Authorised combinations. All other combinations represent Commands not properly Authorised and shall be rejected by a Device.

	authorisingRemotePartyTACellIdentifier		
	RemotePartyRole	KeyUsage	CellUsage
credentialsReplacementMode			
supplierBySupplier	supplier	digitalSignature	management
networkOperatorByNetworkOperator	networkOperator	digitalSignature	management
accessControlBrokerByACB: if authorisingRemotePartyTACellIdentifier is present it must refer to the specified Trust Anchor Cell. If absent and deviceType is not communicationsHubCommunicationsHubFunction, then the mode of accessControlBrokerByACB is valid, otherwise the mode is invalid	accessControlBroker	digitalSignature	management
wanProviderByWanProvider	wanProvider	digitalSignature	management
transCoSByTransCoS	transitionalCoS	digitalSignature	management
supplierByTransCoS	transitionalCoS	digitalSignature	management
anyExceptAbnormalRootByRecovery	recovery	digitalSignature	management
anyByContingency	root	keyCertSign	management

Table 13.3.5.5: Valid credentialsReplacementMode by Remote Party Roles authorising the Command

13.3.5.6 Valid credentialsReplacementMode by the targetTrustAnchorCells specified in the Command

A Command containing a certain credentialsReplacementMode can only validly replace the Security Credentials in a certain subset of Trust Anchor Cells. The Command identifies the Cells that are to have credentials replaced in each targetTrustAnchorCell within each TrustAnchorReplacement in replacements.

Table 13.3.5.6 below lists the only valid targetTrustAnchorCell combinations for each credentialsReplacementMode. All other combinations are invalid. A Command containing any invalid combinations shall not be processed any further by the Device.

	targetTrustAnchorCell		
	RemotePartyRole	KeyUsage	CellUsage
credentialsReplacementMode			
supplierBySupplier	supplier	any valid for GBCS	any valid for GBCS

	targetTrustAnchorCell		
	RemotePartyRole	KeyUsage	CellUsage
credentialsReplacementMode			
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

13.3.5.7 Valid usage of Certificates against the `targetTrustAnchorCells` specified in the Command

[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.]

For each instance of the `TrustAnchorReplacement` structure in the Command, the `keyUsage` in `replacementCertificate` shall be equal to `targetTrustAnchorCell.KeyUsage`. Where this check fails for any one or more of the `TrustAnchorReplacement` instances, the Command shall not be actioned by the Device.

[Note: Save for supplier and network operator roles, each of the 'end entity' Certificates in the Command must have the same `RemotePartyRole` as the Trust Anchor Cell it is to be applied to.]

For each instance of the `TrustAnchorReplacement` structure in the Command where (`targetTrustAnchorCell.RemotePartyRole` <> `supplier`) and (`targetTrustAnchorCell.RemotePartyRole <> networkOperator`), the `RemotePartyRole` in `replacementCertificate` shall be equal to `targetTrustAnchorCell.RemotePartyRole`. Where this check fails for any one or more of the `TrustAnchorReplacement` instances, the Command shall not be actioned by the Device.

Notes:

- mismatches between `RemotePartyRole` in the certificate and the target Trust Anchor Cell are admissible, in relation to CS02b Commands, for `supplier` and `networkOperator` only, and are needed (see Section 4.3.2.5); and

- `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 different Key Pair is not mandated and so validation is not required; any valid supplier Key Agreement certificate may be used in this Trust Anchor Cell.

13.3.5.8 Verifying the CryptographicProtections

In verifying Cryptographic Protections pursuant to this Section 13.3.5.8.1:

- KRP Signature shall, where required by Section 13.3.3.1 for the specified Message Code and `credentialsReplacementMode`, be verified according to the requirements in this Section 13.3.5.8; and
- ACB-SMD MAC, where required by Section 13.3.3.1 for the specified Message Code and `credentialsReplacementMode`, shall be verified according to the requirements in Section 6.2.4.1.2.

If `credentialsReplacementMode` = `anyByContingency` or Message Code = 0x0109 then KRP Signature shall be verified using the public key established according to the requirements of Section 13.3.5.8.1.

If `credentialsReplacementMode` = <> `anyByContingency` or Message Code =<> 0x0109 then KRP Signature shall be verified using the public key identified as per Section 4.3.2.7.2.

If `credentialsReplacementMode` = `accessControlBrokerByACB` or Message Code = 0x0104 and `deviceType` is not `communicationsHubCommunicationsHubFunction` then ACB-SMD MAC shall be verified as per Section 6.2.4.1.2.

13.3.5.8.1 Decrypting the contingency public key and verifying Authorising Remote Party's digital signature against that decrypted key

The Device shall decrypt the Contingency Key that it holds in Trust Anchor Cell {`root, digitalSignature, management`} by undertaking Decryption using the following parameters:

- setting Ciphertext to be encrypted value of the Contingency Key;
- setting Additional Authenticated Data to be 0x31;
- setting the Initialization Vector to be 0xFFFFFFFF0000000000000000; and
- setting the shared symmetric key to be the value in `plaintextSymmetricKey`.

Where Decryption is successful, the Device shall use the Plaintext produced as the Public Key to verify KRP Signature according to the requirements at Section 6.3.4.

The Contingency Key shall have been Encrypted accordingly.

13.3.5.9 Verifying the authenticity of replacement certificates

The Device shall first apply the requirements of Section 12.6 (Device processing of Certificates). If any of those checks fail, the Section 13.3.5.9 check fails.

Where Certification Path Validation is required by this Section 13.3.5.9, the application of time based checks shall be determined as follows:

- if, in the Command, `applyTimeBasedCPVChecks` = `disapply` then time based checks shall NOT be applied by the Device; and
- otherwise time based checks shall be applied or not applied in line with the requirements of Section 4.3.2.8.

If (`credentialsReplacementMode <> anyByContingency`) and (`replacements` does NOT include a `targetTrustAnchorCell` of {`root, keyCertSign, management`}) then the Device shall, for each `replacementCertificate` in `replacements`, undertake Certification Path Validation according to the requirements of Section 4.3.2.8.

If (`credentialsReplacementMode = anyExceptAbnormalRootByRecovery`) and (`replacements` does include a `targetTrustAnchorCell` of {`root, keyCertSign, management`}) 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 each of the other `replacementCertificate` in `replacements`, undertake Certification Path Validation according to the requirements of Section 4.3.2.8. In so doing the Device shall use the details from the `replacementCertificate` in `replacements` specified for updating `{root, keyCertSign, management}` as the root for Certification Path Validation.

If (`credentialsReplacementMode = anyByContingency`) and (`replacements` does include a `targetTrustAnchorCell` of {`root, keyCertSign`}) then the Device shall, for each of the other `replacementCertificate` in `replacements`, undertake Certification Path Validation according to the requirements of Section 4.3.2.8. In so doing the Device shall use the details from the `replacementCertificate` in `replacements` specified for updating {`root, keyCertSign, managment`} as the root for Certification Path Validation.

If any of the above checks fail, the Section 13.3.5.9 check fails.

13.3.5.9.1 Validation of new root Certificate against current root Security Credentials

The Device shall:

- identify the `Certificate` in `replacements` that corresponds to the `targetTrustAnchorCell` of {`root, keyCertSign, management`}. The `Certificate` shall be referred to as `NewWithNew`; then
- identify the `Certificate` in `certificationPathCertificates` that has the same `subjectPublicKey` as the `NewWithNew` Certificate and where `RemotePartyRole` in the `X520OrganisationalUnitName` field = `root`. The Certificate shall be referred to as `NewWithOld`. If no such `Certificate` is found, the Section 13.3.5.9.1 check fails else:

- use the Public Key in the `{root, keyCertSign, management}` Trust Anchor Cell to verify the digital signature in `NewWithOld`. If the digital signature verification fails, the Section 13.3.5.9.1 check fails else:
- use the Public Key in `NewWithNew` to verify the digital signature in `NewWithNew`. If the digital signature verification fails, the Section 13.3.5.9.1 check fails.

13.3.5.10 Required Processing on Change of Remote Party Control

If:

- the `targetTrustAnchorCell` is `{supplier, digitalSignature, management}`; and
- the Entity Identifier in the `targetTrustAnchorCell` is changed by the replacement; and
- the Device is an ESME or a GSME,

then the Device shall:

- set the Supplier Name which it displays to the Common Name in the `subject` field of the `certificate` that was used to populate the `targetTrustAnchorCell;` and
- add an entry in the Billing Data Log with a snapshot cause of 0x00002000 (Change of Supplier) (with the entry added having the same content as is required on Set Payment Mode Or Tariff change);
- reset the Tariff Block Counter Matrix; and
- where the Device is an ESME, end any currently active Boost Period, ALCS [n] Setting Period, APC [n] Limit Period, APC [n] Setting Period and HCALCS [n] Setting Period, where those terms have their SMETS meanings.

If the `targetTrustAnchorCell` is `{root, keyCertSign, management}` and there are any future dated Update Security Credentials or Activate Firmware Commands held on the Device that have not yet executed, and so the `executionDateTime` is in the future, then the Device shall set each `executionDateTime` to '99991231235959Z'.

If the Device is an ESME, GSME or HCALCS, the `targetTrustAnchorCell` is `{supplier, digitalSignature, management}` and there are any future dated Update Security Credentials or Activate Firmware Commands held on the Device that have not yet executed, and so their `executionDateTime` is in the future, then the Device shall set each such `executionDateTime` to '99991231235959Z'.

If the Device is a CHF, the `targetTrustAnchorCell` is `{wANProvider, digitalSignature, management}` and there is any future dated Activate Firmware Command held on the Device that have not yet executed, and so its `executionDateTime` is in the future, then the Device shall set each such `executionDateTime` to '99991231235959Z'.

If the Device is a GPF, the `targetTrustAnchorCell` is `{supplier, digitalSignature, management}` and there are any future dated Update Security Credentials Commands held on the Device that have not yet executed, and so their `executionDateTime` is in the future, then the Device shall set each such `executionDateTime` to '99991231235959Z'.

If:

- the `targetTrustAnchorCell` is `{supplier, digitalSignature, management}` or `{root, keyCertSign, management};` and
- the Entity Identifier in the `targetTrustAnchorCell` is changed by the replacement

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 '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 are pending activation.

Relevant Execution Counters' shall be those which relate to types of Commands:

- which originate from the `remotePartyRole` of this `targetTrustAnchorCell`; and
- whose cryptographic protection is checked using the Public Key in this `targetTrustAnchorCell`.

Where this Update Security Credentials Command contains an `otherRemotePartySeqNumberChange` in which `otherRemotePartyRole` is the `remotePartyRole` of this `targetTrustAnchorCell`, the Device shall set all Relevant Execution Counters to be the corresponding value in `otherRemotePartyFloorSeqNumber`. For clarity, this applies to the `loadController remotePartyRole` when executing a `loadControllerBySupplier` CS02g Update Security Credentials Command.

If:

- `remotePartyRole` of `targetTrustAnchorCell` and that of `authorisingRemotePartyControl` is `supplier`; and
- `keyUsage` of `targetTrustAnchorCell` is `digitalSignature`

then the Device shall set all Relevant Execution Counters to the value in `newRemotePartyFloorSeqNumber`. For clarity, this applies to the Execution Counter for the `loadControllerBySupplier` CS02g Update Security Credentials Command.

If the `deviceType` is `eSME` or `gSME` and (`targetTrustAnchorCell` is `{supplier, digitalSignature, management}`, the Device shall:

- clear all values from the UTRN Counter Cache; and
- 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 whichever of the `newRemotePartyFloorSeqNumber` or the `otherRemotePartyFloorSeqNumber` fields present in the Command.

Note, where `SeqNumber` values are to be used to populate Execution Counters on the Device, these `SeqNumber` values shall be processed as unsigned 64 bit integers, so 8 octets in length. This is because Execution Counters are required to use this representation. Correspondingly, in relation to values in the UTRN Counter Cache (so UTRN Counters), references to the 32 most significant bits of `SeqNumbers` are references to the 32 most significant bits when the value is represented as a 64 bit unsigned integer.

13.3.5.11 The `@UpdateSecurityCredentials.CommandPayload`, `@UpdateSecurityCredentials.ResponsePayload` and `@UpdateSecurityCredentials.AlertPayload` structure definition

Each instance of `@UpdateSecurityCredentials.CommandPayload`, `@UpdateSecurityCredentials.ResponsePayload` and of `@UpdateSecurityCredentials.AlertPayload` shall be an octet string containing the DER encoding of the populated structure defined in this Section 13.3.4, which specifies the structure in ASN.1.

The structure of `Certificate` shall be as defined in ASN.1 in IETF RFC 5912. Note that the `Certificate` structures within IETF RFC 5912 begin after the phrase 'Certificate- and CRL-specific structures begin here'.

```
UpdateSecurityCredentials DEFINITIONS ::= BEGIN

CommandPayload ::= SEQUENCE
{
-- Provide details to allow the Device to identify the Remote Party Role authorising
-- this Command, check whether the rest of the payload is allowable, prevent replay attacks
-- and allow counters / counter caches on the Device to be reset, if the Command changes the Remote Party
-- in control.
-- The Remote Party authorising the Command is that party which generated the KRP Signature (or the Access Control Broker
-- if there is no KRP Signature)

authorisingRemotePartyControl                 AuthorisingRemotePartyControl,

-- One TrustAnchorReplacement structure is required for each Trust Anchor Cell that is to be updated

replacements                                  SEQUENCE OF TrustAnchorReplacement,

-- 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 the root public
```

```
-- key as currently stored on the Device.

certificationPathCertificates                 SEQUENCE OF Certificate,

-- If the Command is to be future dated, specify the date-time at which the certificate replacement is to happen

executionDateTime                             GeneralizedTime OPTIONAL

}

ResponsePayload ::=                           SEQUENCE
{
   -- if the Command is future dated, the Response will not have any details of execution (those will be in the subsequent alert)

   commandAccepted                            NULL,

   -- if the Command is for immediate execution, the Response will detail the outcomes

   executionOutcome                           ExecutionOutcome OPTIONAL

}

AlertPayload ::=                              SEQUENCE
{
   -- specify the Alert Code
   alertCode                                  INTEGER(0..4294967295),

   -- specify the date-time of execution
   executionDateTime                          GeneralizedTime,


   -- detail what happened when the future dated Command was executed

   executionOutcome                           ExecutionOutcome

}

ExecutionOutcome ::=                          SEQUENCE
{
-- Provide details of the corresponding Command that may not be in the standard GBCS message header. Specifically the
-- mode in which the Command was invoked, the Originator Counter in the original Command and the resulting changes to any
-- replay counters held on the Device

authorisingRemotePartySeqNumber               SeqNumber,
```

```
credentialsReplacementMode                    CredentialsReplacementMode,
remotePartySeqNumberChanges                   SEQUENCE OF RemotePartySeqNumberChange,

-- For each replacement in the Command, detail the outcome and impacted parties

replacementOutcomes                           SEQUENCE OF ReplacementOutcome

}

AuthorisingRemotePartyControl ::=             SEQUENCE
{
-- Specify the replacement mode so that the Device can check that the Remote Party Role is allowed to
-- authorise this type of replacement and that all replacements in the payload are allowed within this
-- replacement mode

credentialsReplacementMode                    CredentialsReplacementMode,

-- Only if credentialsReplacementMode = anyByContingency, provide the symmetric key to decrypt
-- the Contingency Public Key in the (root, digitalSignature, management) Trust Anchor Cell

plaintextSymmetricKey                         [0] IMPLICIT OCTET STRING OPTIONAL,

-- Specify whether the time based checks as part of any Certificate Path Validation should be applied

applyTimeBasedCPVChecks                       [1] IMPLICIT INTEGER {apply(0), disapply(1)} DEFAULT apply,

-- Identify which of the Public Keys on the Device is to be used in checking KRP Signature
-- 'authorisingRemotePartyTACellIdentifier' may only be omitted when
-- the access control broker is updating its own credentials and the target device is not a CHF.
-- In all other cases it is mandatory.

authorisingRemotePartyTACellIdentifier        [2] IMPLICIT TrustAnchorCellIdentifier OPTIONAL,

-- 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.

authorisingRemotePartySeqNumber               [3] IMPLICIT SeqNumber,

-- If the Command is to effect a change of control, then newRemotePartyFloorSeqNumber must be included
-- and will be the value used to prevent replay of Update Security Credentials Commands for the
-- new controlling Remote Party.

newRemotePartyFloorSeqNumber                  [4] IMPLICIT SeqNumber OPTIONAL,
```

```
-- Some Commands on the Device may use a different Originator Counter sequence for Protection Against Replay. At this
-- version of the GBCS, 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.

newRemotePartySpecialistFloorSeqNumber        [5] IMPLICIT SEQUENCE OF SpecialistSeqNumber OPTIONAL,

-- In some cases, one party acting in one Remote Party Role may be replacing certificates for a different Remote Party Role.
-- In some cases, Execution Counters need also to be reset for those other Remote Party Role(s)

otherRemotePartySeqNumberChanges              [6] IMPLICIT SEQUENCE OF RemotePartySeqNumberChange OPTIONAL
}

RemotePartySeqNumberChange ::=                 SEQUENCE
{
otherRemotePartyRole                          RemotePartyRole,
otherRemotePartyFloorSeqNumber                SeqNumber,
newRemotePartySpecialistFloorSeqNumber        SEQUENCE OF SpecialistSeqNumber OPTIONAL
}

SpecialistSeqNumber ::=                       SEQUENCE
{
-- Specify the usage of the SeqNumber
seqNumberUsage                                SeqNumberUsage,

-- Specify the associated SeqNumber
seqNumber                                     SeqNumber
}

SeqNumberUsage ::=                            INTEGER
{
-- Define the full set of discrete usages on a Device. The only specialist
-- counter is for Prepayment Top Up (which is set independently of other counters). This may only be
-- included when changing Supplier Security Credentials on an ESME or GSME.

prepaymentTopUp                               (0)
}

SeqNumber ::=                                 INTEGER (0.. 18446744073709551615)


TrustAnchorReplacement ::=                    SEQUENCE
{
-- Provide the new end entity certificate
```

```
replacementCertificate                        Certificate,

-- Specify where it is to go (specifically which Trust Anchor Cell is to have its details replaced using
-- the new end entity certificate)

targetTrustAnchorCell                         TrustAnchorCellIdentifier
}


ReplacementOutcome ::=                        SEQUENCE
{
affectedTrustAnchorCell                       TrustAnchorCellIdentifier,
statusCode                                    StatusCode,

-- The GBCS Certificate requirements mean that the Subject Unique ID attribute in the subject field of a certificate will always
-- contain the 64 bit unique number that equates to Entity Identifier. existingSubjectUniqueID should be set
-- accordingly based on the contents of the Trust Anchor Cell prior to Command processing.

existingSubjectUniqueID                       OCTET STRING,

-- The GBCS Certificate requirements mean that subjectKeyIdentifier attributes will all be 8 byte SHA-1 Hashes.
-- existingSubjectKeyIdentifier should be set accordingly based on the contents of the Trust Anchor Cell prior to
-- Command processing.

existingSubjectKeyIdentifier                  OCTET STRING,

-- The Subject Unique ID in the subject field of the certificate in this TrustAnchorReplacement

replacingSubjectUniqueID                      OCTET STRING,

-- The subjectKeyIdentifier in the certificate in this TrustAnchorReplacement

replacingSubjectKeyIdentifier                 OCTET STRING
}

TrustAnchorCellIdentifier ::=                 SEQUENCE
{
-- Which Remote Party Role does this Cell relate to?

trustAnchorCellRemotePartyRole                RemotePartyRole,

-- To what cryptographic use can the Public Key in this Cell be put? Some Remote Party Roles
-- (e.g. supplier) can have more than one Public Key on a Device and each one would only have
-- a single cryptographic use.
```

```
trustAnchorCellKeyUsage                       KeyUsage,

-- trustAnchorCellUsage is to allow for multiple Public Keys of the same keyUsage for the same Remote
-- Party Role. It will be absent except where used to refer to the Supplier Key
-- Agreement Key used solely in relation to validating Supplier generated MACs on Prepayment Top Up
-- transactions

trustAnchorCellUsage                          CellUsage DEFAULT management
}

CellUsage ::=                                 INTEGER {management(0), prePaymentTopUp(1)}

RemotePartyRole ::=                           INTEGER
{
-- Define the set of Remote Party Roles in relation to which a Device may need to undertake
-- processing. Note that most Devices will only support a subset of these.

root                                          (0),
recovery                                      (1),
supplier                                      (2),
networkOperator                               (3),
accessControlBroker                           (4),
transitionalCoS                               (5),
wanProvider                                   (6),
issuingAuthority                              (7),    -- Devices will receive such Certificates but they do not need to store
                                                      -- them over an extended period

-- The 'other' RemotePartyRole is for a party whose role does not allow it to invoke any Device function apart from
-- UpdateSecurityCredentials. This is to allow for Device functionality to be locked out of usage until a valid
-- Remote Party can be identified e.g. where roles cannot be fixed until a Device is brought in to operation

other                                         (127)

}

-- KeyUsage is only repeated here for clarity. It is defined in RFC 5912

KeyUsage ::=                                  BIT STRING
{
-- Define valid uses of Public Keys held by Devices in their Trust Anchor Cells.

digitalSignature                              (0),
contentCommitment                             (1),    -- not valid for GBCS compliant transactions
```

```
keyEncipherment                                (2),    -- not valid for GBCS compliant transactions
dataEncipherment                               (3),    -- not valid for GBCS compliant transactions
keyAgreement                                   (4),
keyCertSign                                    (5),
cRLSign                                        (6),
encipherOnly                                   (7),    -- not valid for GBCS compliant transactions
decipherOnly                                   (8)     -- not valid for GBCS compliant transactions
}

CredentialsReplacementMode ::=                 INTEGER
{
-- Define the valid combinations as to which Remote Party Roles can replace which kinds of Trust Anchors.

-- Normal operational replacement modes
supplierBySupplier                             (2),
networkOperatorByNetworkOperator               (3),
accessControlBrokerByACB                       (4),
wanProviderByWanProvider                       (5),
transCoSByTransCoS                             (6),
supplierByTransCoS                             (7),

-- Recovery modes
anyExceptAbnormalRootByRecovery                (8),
anyByContingency                               (9)
}

-- The GBCS only allows for a constrained set of Trust Anchor Cell operations and so the list of possible outcomes
-- is more limited than in RFC 5934. The list below is that more constrained subset

StatusCode ::=                                 ENUMERATED {

success                                        (0),

-- badCertificate is used to indicate that the syntax for one or more certificates is invalid.

badCertificate                                 (5),

-- noTrustAnchor is used to indicate that the authorityKeyIdentifier does not identify the public key of a
-- trust anchor or a certification path that terminates with an installed trust anchor

noTrustAnchor                                  (10),

-- insufficientMemory indicates that the update could not be processed because the Device did not
-- have sufficient memory
```

```
insufficientMemory                          (17),

-- resourcesBusy indicates that the resources necessary to process the replacement are not available at the
-- present time, but the resources might be available at some point in the future.

resourcesBusy                               (30),

-- other indicates that the update could not be processed, but the reason is not covered by any of the assigned
-- status codes.  Use of this status code SHOULD be avoided.

other                                       (127) }

END
```

13.3.5.12 Requirements for `AuthorisingRemotePartyControl` elements – informative

All bar two parts of the `AuthorisingRemotePartyControl` structure are optional. This Section summarises when each of the optional 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 Key Agreement credentials (in which case it should be omitted)
`authorisingRemotePartySeqNumber`	Always required
`newRemotePartyFloorSeqNumber`	If the Command is to effect a change of control, then `newRemotePartyFloorSeqNumber` 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.

`AuthorisingRemotePartyControl` element	Notes
`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 Execution Counters for that other role or roles

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
Service 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:

1. 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:
    - those held in the {`wANProvider, digitalSignature, management`} Trust Anchor Cell, if the target Device's `deviceType` equals `communicationsHubCommunicationsHubFunction`;

- those held in the {`supplier, digitalSignature, management`} Trust Anchor Cell, if the target Device's `deviceType` does not equal `communicationsHubCommunicationsHubFunction`;

2. validate that the value of `keyUsage` in `CommandPayload` is either `digitalSignature` only or `keyAgreement` only. If this validation fails then the Device shall set `issueCredentialsResponseCode` to `invalidKeyUsage`, and process from step 6;
3. generate a Private-Public Key Pair and store the Private Key so generated in the Pending Private Key Cell determined by the value of `keyUsage` in `CommandPayload`. If the step fails then the Device shall set `issueCredentialsResponseCode` to `keyPairGenerationFailed`, and process from step 6;
4. with the ASN.1 terms in this step (that are not defined in this Section 13.4.4) having the meaning of IETF RFC 2986[36]; generate a `CertificationRequest` which:
   - complies with the requirements of IETF RFC2986 and IETF RFC 5912;
   - is DER encoded, in line with the recommendation of IETF RFC 5967[37];
   - has `subjectPublicKey` set to the bit string representation of the Public Key generated in step 3;
   - incorporates an `extensionRequest` identified by `id-ce-keyUsage` which shall contain the `keyUsage` value specified in `CommandPayload`;
   - incorporates an `extensionRequest` identified by `id-ce-subjectAltName` which shall contain a single `GeneralName` of type `OtherName` that is further sub-typed as a `HardwareModuleName` (`id-on-HardwareModuleName`) as defined in IETF RFC 4108. The `hwSerialNum` field shall be set to the Device's Entity Identifier; and
   - has the `signature` generated using the Private Key generated in step 3;
5. if the generation of `CertificationRequest` is not successful then the Device shall set `issueCredentialsResponseCode` to `cRProductionFailed`; and
6. create a Response according to the requirements of Section 13.4.7, apply the Response Cryptographic Protection required for a Response of Message Category SME.C.C, and send the Response.

13.4.5 Response Processing

Response Recipient Verification may be undertaken as specified in this GBCS for a Response of Message Category SME.C.C. The `issueCredentialsResponseCode` field, where present in the Response, may be decoded according to the ASN.1 definitions at Section 13.4.6.

[36] https://tools.ietf.org/html/rfc2986
[37] https://tools.ietf.org/html/rfc5967

13.4.6 Issue Security Credentials Command and Response payloads – structure definition

Each instance of `@IssueSecurityCredentials.CommandPayload` and of `@IssueSecurityCredentials.ResponsePayload` shall be 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 notation.

```
IssueSecurityCredentials DEFINITIONS ::= BEGIN

CommandPayload ::=                              SEQUENCE
{
     -- specify the keyUsage to which the generated key-pair will be put, if subsequently authorised
     keyUsage                                   KeyUsage

}

ResponsePayload ::=                             CHOICE

{
     -- if the Command was successful, provide the generated Certification Request. CertificationRequest shall
     -- be as defined in ASN.1 by IETF RFC 5912.  For reference, it is in the section headed 'ASN.1 Module for RFC 2986'
     certificationRequest                       CertificationRequest,

     -- if the Command was unsuccessful, detail the failure

     issueCredentialsResponseCode               IssueCredentialsResponseCode
}

-- KeyUsage is only repeated here for ease of reference. It is defined in IETF RFC 5912

KeyUsage ::=                                BIT STRING
{
-- Define valid uses of Public Keys held by Devices in their Trust Anchor Cells.

digitalSignature                            (0),
contentCommitment                           (1),    -- not valid for GBCS compliant transactions
keyEncipherment                             (2),    -- not valid for GBCS compliant transactions
dataEncipherment                            (3),    -- not valid for GBCS compliant transactions
keyAgreement                                (4),
keyCertSign                                 (5),    -- not valid for this Use Case
cRLSign                                     (6),    -- not valid for this Use Case
encipherOnly                                (7),    -- not valid for GBCS compliant transactions
decipherOnly                                (8)     -- not valid for GBCS compliant transactions
}
```

```
IssueCredentialsResponseCode::=                    INTEGER
{
     invalidKeyUsage                               (1),
     keyPairGenerationFailed                       (2),
     cRProductionFailed                            (3)
}


END
```

13.4.7 Constructing the `@IssueSecurityCredentials.CommandPayload` and of `@IssueSecurityCredentials.ResponsePayload`

`@IssueSecurityCredentials.CommandPayload` shall have the structure defined in Section 13.4.6, and the Remote Party constructing 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

Table 13.4.7a: `@IssueSecurityCredentials.CommandPayload` population

`@IssueSecurityCredentials.ResponsePayload` shall have the structure defined in Section 13.4.6, and the Device constructing the Response shall populate with values according to Table 13.4.7b.

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`			

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
`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

Table 13.4.7b: `@IssueSecurityCredentials.ResponsePayload` population

13.5 CS02d Update Device Certificates on Device

13.5.1 Description

This Section 13.5 covers the creation, validation and processing of (i) Update Device Certificates on Device, Commands and (ii) Responses to such Commands.

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
Service 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

13.5.3 Construction of Commands

Update Device Certificate on Device Command Payloads shall be constructed as specified in Section 13.5.7, and Cryptographic Protection I and Cryptographic Protection II shall be applied as required for a Command of Message Category SME.C.C.

13.5.4 Device processing of Commands and Response handling

The Device receiving an Update Device Certificate on Device Command shall undertake processing steps in the sequence defined in this Section 13.5.4.

In processing an Update Device Certificate on Device Command, the Device shall:

1. undertake Command Authenticity and Integrity Verification as required for a Command of Message Category SME.C.C, except that check 4 in Section 6.2.4.1.1 may be undertaken after the checks in Section 6.2.4.1.2. The Security Credentials used to verify Cryptographic Protection I shall be:
    - those held in the {`wANProvider, digitalSignature, management`} Trust Anchor Cell, if the target Device's `deviceType` equals `communicationsHubCommunicationsHubFunction`; or

- those held in the {supplier, digitalSignature, management} Trust Anchor Cell, if the target Device's deviceType does not equal communicationsHubCommunicationsHubFunction.

2. establish the values of keyUsage, subjectPublicKey and hwSerialNum in certificate in the CommandPayload. If any of the values cannot be established then the Device shall set updateDeviceCertResponseCode to invalidCertificate, and process from step 10;
3. validate that hwSerialNum established at step 2 is the Device's Entity Identifier. If this validation fails then the Device shall set updateDeviceCertResponseCode to wrongDeviceIdentity, and process from step 10;
4. validate that keyUsage established at step 2 is either digitalSignature only or keyAgreement only. If this validation fails then the Device shall set updateDeviceCertResponseCode to invalidKeyUsage, and process from step 10;
5. validate that the Device holds a Pending Private Key for the keyUsage as established at step 2. If this validation fails then the Device shall set updateDeviceCertResponseCode to noCorrespondingKeyPairGenerated, and process from step 10;
6. validate that subjectPublicKey established at step 2 is the bit string representation of the Public Key corresponding to the Pending Private Key identified at step 5. If this validation fails then the Device shall set updateDeviceCertResponseCode to wrongPublicKey, and process from step 10;
7. store certificate. If this step fails then the Device shall set updateDeviceCertResponseCode to certificateStorageFailed, and process from step 10;
8. set the Current Private Key to have the value of the Pending Private Key for the keyUsage established at step 2. If this step fails then the Device shall set updateDeviceCertResponseCode to privateKeyChangeFailed, and process from step 10;
9. set updateDeviceCertResponseCode to success; and
10. create a Response according to the requirements of Section 13.5.7, apply the Response Cryptographic Protection required for a Response of Message Category SME.C.C, and send the Response.

If all steps were successful and this was a change of digitalSignature certificate, the Response shall be signed using the private key corresponding to the new certificate. If there was a failure, the Response shall be signed using the private key corresponding to the pre-existing key pair.

Once the Pending Private Key becomes the Current Private Key, the Device will be using the new Private Key and this will affect all Remote Parties interacting with the Device; specifically they will need to use the new Certificate corresponding to the Private Key now in use.

13.5.5 Response Processing

Response Recipient Verification may be undertaken as specified in this GBCS for a Response of the relevant Message Category. The updateDeviceCertResponseCode field may be decoded according to the ASN.1 definitions at Section 13.5.6.

If this was a change of `digitalSignature` certificate, the public key to be used to verify the Device's signature is dependent on the value of `updateDeviceCertResponseCode`. If `updateDeviceCertResponseCode` is `success` then the Private Key used for Signing will have changed. If `updateDeviceCertResponseCode` is other than `success`, the 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.

```
UpdateDeviceCertificateonDevice DEFINITIONS ::= BEGIN

CommandPayload ::=                              Certificate
     -- provide the certificate which the Device is to store
     -- the ASN.1 specification of certificate shall be as defined in IETF RFC 5912 for IETF RFC 5280

ResponsePayload ::=                             UpdateDeviceCertResponseCode

     -- if the Command was unsuccessful, detail the failure; otherwise respond with success

UpdateDeviceCertResponseCode::=                 INTEGER
{
     success                                    (0),
     invalidCertificate                         (1),
     wrongDeviceIdentity                        (2),
     invalidKeyUsage                            (3),
     noCorrespondingKeyPairGenerated            (4),
     wrongPublicKey                             (5),
     certificateStorageFailed                   (6),
     privateKeyChangeFailed                     (7)
}

END
```

13.5.7 Constructing the `@UpdateDeviceCertificateonDevice.CommandPayload` and `@UpdateDeviceCertificateonDevice.ResponsePayload`

`@UpdateDeviceCertificateonDevice.CommandPayload` shall have the structure defined in Section 13.5.6, and the Remote Party constructing the Command shall populate with values according to Table 13.5.7a.

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
`@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

`@UpdateDeviceCertificateonDevice.ResponsePayload` shall have the structure defined in Section 13.5.6, and the Device constructing the Response shall populate with 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	Notes
`@UpdateDeviceCertificateonDevice.ResponsePayload`				
`UpdateDeviceCertResponseCode`	`INTEGER`	Shall be populated according to the processing defined in Section 13.5.4	Mandatory	

Table 13.5.7b: `@UpdateDeviceCertificateonDevice.ResponsePayload` population

13.6 CS02e 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
Service 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 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 13.6.2: Use Case Cross References for Provide Device Certificates from Device Command and Response

13.6.3 Construction of Commands

Provide Device Certificates from Device Command Payloads shall be constructed as specified in Section 13.6.7 and Cryptographic Protection I and Cryptographic Protection II shall be applied as required for a Command of Message Category SME.C.C.

13.6.4 Device processing of Commands and Response handling

The Device receiving a Provide Device Certificates from Device Command shall undertake processing steps in the sequence defined in this Section 13.6.4.

In processing a Provide Device Certificates from Device Command, the Device shall:

1. undertake Command Authenticity and Integrity Verification as required for a Command of Message Category SME.C.C, except that Cryptographic Protection II shall not be verified. The Security Credentials used to verify Cryptographic Protection I shall be:
   - those held in the {`wANProvider, digitalSignature, management`} Trust Anchor Cell, if the target Device's `deviceType` equals `communicationsHubCommunicationsHubFunction`; or

- those held in the {`supplier, digitalSignature, management`} Trust Anchor Cell, if the target Device's `deviceType` does not equal `communicationsHubCommunicationsHubFunction;`

2. validate that `keyUsage` in `CommandPayload` is either `digitalSignature` only or `keyAgreement` only. If this validation fails then the Device shall set `provideDeviceCertResponseCode` to `invalidKeyUsage`, and process from step 5;
3. confirm that the Device holds a certificate which (1) is for the `keyUsage` identified at step 2, (2) contains in `hwSerialNum` a value equal to the Device's Entity Identifier and (3) contains in `subjectPublicKey` the bit string representation of the Public Key corresponding to the Current Private Key for this `keyUsage`. If this validation fails then the Device shall set `provideDeviceCertResponseCode` to `noCertificateHeld`, and process from step 5;
4. retrieve the certificate identified in step 3. If this step fails then the Device shall set `provideDeviceCertResponseCode` to `certificateRetrievalFailure`, and process from step 5; and
5. create a Response according to the requirements of Section 13.6.7, apply the Response Cryptographic Protection required for a Response of Message Category SME.C.C, and send the Response.

13.6.5 Response Processing

Response Recipient Verification may be undertaken as specified in this GBCS for a Response of the Message Category SME.C.C. The `provideDeviceCertResponseCode` field may be decoded according to the ASN.1 definitions at Section 13.6.6.

13.6.6 Provide Device Certificates from Device Command and Response payloads – structure definition

Each instance of `@ProvideDeviceCertificateFromDevice.CommandPayload` and of `@ProvideDeviceCertificateFromDevice.ResponsePayload` shall be an octet string containing the DER encoding of the populated structure defined in this Section 13.6.6 which specifies the structure in ASN.1 notation.

```
ProvideDeviceCertificateFromDevice DEFINITIONS ::= BEGIN

CommandPayload ::=                              SEQUENCE
{
     -- specify the KeyUsage of the Certificate to be returned
     keyUsage                                   KeyUsage

}

ResponsePayload ::=                             CHOICE

{
     -- if the Command was successful, provide the certificate
     certificate                                Certificate,

     -- if the Command was unsuccessful, detail the failure

     provideDeviceCertResponseCode              ProvideDeviceCertResponseCode
}


-- KeyUsage is only repeated here for ease of reference. It is defined in RFC 5912

KeyUsage ::=                                BIT STRING
{
-- Define valid uses of Public Keys held by Devices in their Trust Anchor Cells.

digitalSignature                            (0),
contentCommitment                           (1),    -- not valid for GBCS compliant transactions
keyEncipherment                             (2),    -- not valid for GBCS compliant transactions
dataEncipherment                            (3),    -- not valid for GBCS compliant transactions
keyAgreement                                (4),
keyCertSign                                 (5),    -- not valid for this Use Case
cRLSign                                     (6),    -- not valid for this Use Case
encipherOnly                                (7),    -- not valid for GBCS compliant transactions
```

```
decipherOnly                                (8)     -- not valid for GBCS compliant transactions
}

ProvideDeviceCertResponseCode::=                INTEGER
{
     invalidKeyUsage                            (1),
     noCertificateHeld                          (2),
     certificateRetrievalFailure                (3)
}


END
```

13.6.7 Constructing the `@ProvideDeviceCertificateFromDevice.CommandPayload` and `@ProvideDeviceCertificateFromDevice.ResponsePayload`

`@ProvideDeviceCertificateFromDevice.CommandPayload` shall have the structure defined in Section 13.6.6 and the Remote Party 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

Table 13.6.7a: `@ProvideDeviceCertificateFromDevice.CommandPayload` population

`@ProvideDeviceCertificateFromDevice.ResponsePayload` shall have the structure defined in Section 13.6.6, and the Device constructing the Response shall populate with values according to Table 13.6.7b.

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 13.6.4	Mandatory	Mandatory if certificate is not successfully produced

Table 13.6.7b: `@ProvideDeviceCertificateFromDevice.ResponsePayload` population

13.7 Pair-wise Authorisation of Devices

13.7.1 Introduction to pair-wise Authorisation of Devices – informative

13.7.1.1 The role of pair-wise Authorisation – informative

This Section 13.7 includes the Use Cases related to the Authorisation (and the removal of Authorisation) for pair-wise, secure application layer interaction between two Devices on the same SMHAN. It also covers the related Use Cases for backing up and restoring the GPF Device Log.

The process of authorising two Devices to communicate is referred to as 'Joining'[38]. Removal of such authorisation is referred to as 'Un-joining'. Correspondingly, Remote Party Commands are specified in this Section 13.7 for instructing Devices that they are to 'Join' or 'Unjoin'.

In line with the SMETS and CHTS Device Log requirements, two Devices on a HAN must only be capable of interacting at the application layer if they are currently Joined. They must not be capable of interacting if (1) they have never been Joined or (2) they have been Unjoined.

The application layer interactions between Devices on the same SMHAN must conform to the Device Based Access Controls (DBAC) as defined in Section 13.7.3. For example, an ESME must not be capable of processing an 'Enable Supply' Command from an IHD or an HCALCS.

It is a precondition of Joining that both Devices have been 'White-listed' on to the HAN (as per Use Case 'CCS01 Add Device to CHF Device log') so that they are able to communicate over the HAN at the network layer (and so have network access). The GPF may be configured to be in the CHF's Device Log at manufacture. A Device on a white-list can be removed from the white-list. It must then be unable to communicate over the HAN, and so unable to interact at the application layer with any Devices to which it was 'Joined'.

In SMETS terminology:

- the CHF's Device Log holds the list of currently white-listed Devices on the SMHAN; and
- the Device Log on an ESME, SAPC, GSME, GPF, HCALCS or PPMID Device holds the Entity Identifiers, Device Types and related Security Credentials of other Devices on the HAN to which the Device is currently Joined (and so Authorised to interact with at an application layer).

The process of white-listing a Device, and its subsequently obtaining network access, establishes a shared secret key between the Device and the Communications Hub. The Gas Proxy Function, which is part of the Communications Hub, uses this shared secret key, combined with a Device being entered in to its Device Log, for application layer authorisation.

IHDs and other Type 2 Devices are not required to have a Device Log (as defined in SMETS). They are required to store security and related details of the Devices to which they are Joined as required by ZSE however (otherwise they would be cryptographically unable to understand the information being sent to them by the Joined Devices).

IHDs and other Type 2 Devices can only read application layer information from Devices to which they are Joined (either by requesting the information from the Device or by receiving

[38] This is unrelated to the ZSE meaning of 'joining'

information published by the Device). When a PPMID is joined to a GPF, the PPMID can only read information from the GPF to which it is Joined.

When other types of Device are Joined (e.g. HCALCS, PPMID), they can also exchange Commands and Responses at the application layer. For example, an ESME that is Joined to an HCALCS can send a Command to the HCALCS to turn its switch on and the HCALCS can send a Response saying whether it has done that. A PPMID can send an 'enable supply' Command to an ESME etc.

13.7.1.2 The joining sequence – informative

Statements in this Section 13.7.1.2 apply to an SAPC as if it were an ESME.

There are three types of Join:

- Join Method B: this is a Join involving a Type 2 Device or a GPF;
- Join Method C: this is a Join between a GSME and a PPMID; and
- Join Method A: this is any Join which is not covered by Method B or C.

Except for Method C, all Joins use the ZSE cryptography which requires exchange of messages between the two Devices to establish the shared secret that the two Devices will need to use. Method C uses the cryptography of Section 4 of this GBCS.

Only certain combinations of Devices can be validly 'Joined'. Table 13.7.1.2 summarises valid combinations:

Device Name		ESME	GSME	Comms Hub (CHF)	Comms Hub (GPF)	HCALCS	PPMID	Type 2 (IHD or CAD)
	`deviceType`	1	0	2	3	4	5	6
ESME	1	Not permitted						
GSME	0	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

Table 13.7.1.2: Permitted Joins

A Method A Join always involves an ESME and therefore any HAN exchanges required by a Method A Join shall always be instigated by the ESME involved. In this context the ESME is referred to as the `methodAInitiator`, since it initiates Method A Joins.

The additional step with a Method A Join is that the other Device must first be sent a Join Command detailing the ESME with which it is allowed to Join. On receipt, the Device should add the ESME details to its Device Log and send a Response accordingly. If, subsequently,

the Device is asked to undertake key establishment, it must check that the requesting Device is in its Device Log.

Only one Device in a Method B Join is remotely instructed. Thus, the HAN exchanges required by a Method B Join shall always be instigated by the Device receiving such a Command. From Table 13.7.1.2, this is always a GSME or ESME except where a GPF is to Join to a PPMID, IHD or CAD. Thus, the sequence of a Method B Join is that the ESME / GSME / GPF:

- is sent a Join Command containing the Entity Identifier of the Device to which it is to Join and that other Device's `DeviceType`;
- verifies the cryptographic protection on the Command and checks to make sure it is well formed and valid;
- updates its Device Log to include details of the new Device;
- for an ESME[39], undertakes the key establishment process with the specified Device, as per the ZSE specification; and
- creates and sends a Response detailing the success or otherwise of its actions.

A Method C join does not require exchange of Messages between the two Devices for the establishment of the shared secret. Thus the sequence of a Method C Join is that each of the GSME and PPMID:

- is sent a Join Command containing the Entity Identifier of the Device to which it is to Join, that other Device's `DeviceType` and Key Agreement `Certificate`;
- verifies the cryptographic protection on the Command and does checks to make sure it is well formed and valid;
- updates its Device Log to include details of the new Device;
- checks there is a well-formed Device Certificate in the Command;
- optionally calculates the shared secret using the Device Certificate of the other Device (which is provided in the Command); and
- creates and sends a Response detailing the success or otherwise of its actions.

13.7.1.3 The format of Message Payloads – informative

In common with other GBCS Remote Messages related to the management of Security Credentials, the payloads of Commands and Responses defined in this Section 13.7.1.3 are specified using ASN.1, with DER encoding to be applied to Command and Response payloads.

13.7.2 Device Requirements

All Devices shall:

- support the ZSE Key Establishment Cluster as specified in Annex C of the ZSE cluster;
- support 'Crypto Suite 2' as defined in the ZSE specification; and
- use 'Crypto Suite 2' when undertaking any associated Key Establishment process.

[39] 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.

CHF, ESME, SAPC, HCALCS, PPMID and IHD shall support the mechanism at ZSE 5.4.7.4. A key that is provided to a pair of Devices by the CHF through that ZSE 5.4.7.4 mechanism shall, in this Section 13, be referred to as a 'Partner Link Key'.

An ESME shall be configured to be a ZSE 'Router', as defined in ZSE so that communications between the ESME and Devices Joined to the ESME, that are not operating on a Sub GHz Channel, are not reliant on availability of the Communications Hub.

Where an SAPC is not operating on a Sub GHz Channel, the SAPC shall be configured to be a ZSE 'Router', as defined in ZSE so that communications between the SAPC and Devices Joined to the SAPC, that are not operating on a Sub GHz Channel, are not reliant on availability of the Communications Hub.

Pursuant to the requirements in the SMETS and the CHTS requirement, Devices shall only communicate at an application layer with other Devices that are currently in their Device Log and are permitted by Device Based Access Controls (DBAC) as defined at Section 13.7.3. Such communications shall always be secured using the shared secrets established pursuant to Sections 13.7.4.

Application layer communications within the scope of the DBAC requirement are HAN Only Messages, including provision of information to a PPMID or Type 2 Device. Note that HAN Only Messages between a PPMID and GSME have a structure that is specified in this GBCS in the corresponding Use Cases, and those relate only to Add Credit and Activate Emergency Credit Commands and the corresponding Responses.

Access to the *MeteringDeviceType* attribute in the *Metering* Cluster shall not be within the scope of the DBAC requirements in Section 13.7.3, so as to facilitate use of the Partner Link Key mechanism as a pre-requisite to Joins involving an ESME or an SAPC.

Each entry in a non CHF Device Log shall contain the Entity Identifier of the Authorised Device and its `deviceType`.

The Entity Identifier of a Device with `DeviceType` of `communicationsHubGasProxyFunction` shall be the EUI 64-bit identifier of the ZigBee radio interface installed in the Communications Hub.

13.7.3 Device Based Access Control

In relation to information and functionality within SMETS, a Device shall, when it is a recipient of a Command or request for information from another Device on its SMHAN, only attempt to action that Command when:

- the sending Device's Entity Identifier is in the recipient Device's Device Log;
- the ZSE cryptographic protection on the Message is authenticated using the Shared Secret / Shared Secret Key established with the sending Device; and
- the Command or request for information is explicitly allowed by a cell in Tables 13.7.3a and 13.7.3b, in terms of the `DeviceType` of the sending (client) and receiving (service) Device. The receiving Device shall determine the sending Device's `DeviceType` by reference to its Device Log entry for that sending Device.

Where a Device is a recipient of a Command or request for information from another Device on its SMHAN that does not meet the access requirements of this Section 13.7.3, it shall:

- generate an entry in the Security Log recording failed Authentication;
- discard the Command or request for information without execution and without sending a Response; and

- send an Alert notifying the failed Authentication, constructed as specified in Section 6.2.4.2, populated with the relevant Alert Code from Section 16, to the Known Remote Party specified in Table 16.2.

An ESME and an SAPC shall not action any ZSE Local Change Supply command from a PPMID where Proposed Supply Status has any value other than 0x02 ('Supply ON').

For SMETS section 7.5.4.1 and 7.5.4.2 interactions, the PPMID and GSME shall comply with Use Cases PCS01 and PCS02 respectively.

Device Name	Server / recipient	ESME / SAPC	GSME	Comms Hub (GPF)	HCALCS	PPMID	Type 2 (IHD or CAD)
Client / sender	`deviceType`	1	0	3	4	5	6
ESME / SAPC	1	-	-	-	5.6.4.1 / 9.5.3.1	-	-
GSME	0	-	-	-	-	-	-
Comms Hub (GPF)	3	-	Request for Information	-	-	-	-
HCALCS	4	8.5.2.1	-	-	-	-	-
PPMID	5	7.5.5.1[40] 7.5.5.2[41] 7.5.5.3[42] Request for Information	7.5.4.1 7.5.4.2	Request for Information	-	-	-
Type 2 (IHD or CAD)	6	Request for Information	-	Request for Information	-	-	-

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 / 8.5.1.2 / 9.5.3.1	Control HCALCS [n] / Control HCALCS	Load Control Event
5.6.3.21 / 8.5.2.1 / 9.5.2.12	Request Control of HCALCS [n] / Request Control of HCALCS	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

Requirements:

[40] For an SAPC, this shall only be allowed by the SAPC where the SAPC supports the corresponding SMETS functionality.
[41] For an SAPC, this shall only be allowed by the SAPC where the SAPC supports the corresponding SMETS functionality.
[42] For an SAPC, this shall only be allowed by the SAPC where the SAPC supports the corresponding SMETS functionality.

- in this Section 13.7.4 and in all sub Sections of it;
- in Use Case CS03A1 Method A Join (Meter);
- in Use Case CS03B Method B Join;
- in Use Case CS04AC Method A or C Unjoin;
- in Use Case CS04B Method B Unjoin; and
- in Use Case CS07 Read Device Join Details;

shall apply to an SAPC as if it were an ESME, and as if its `deviceType` were `eSME`.

Formatted: Font color: Auto

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 13.7.4.1[43]. The Service 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 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	Access Control Broker where the Command is addressed to a PPMID; Supplier otherwise

0x000E	CS03B Method B Join	`gSME`
		`eSME`
		`communicationsHubGasProxyFunction`
			Join Device	SME.C.NC	Supplier, Access Control Broker

0x00AF	CS03C Method C Join	`gSME` `type1PrepaymentInterfaceDevice`	Join Device	SME.C.C	Access Control Broker where the Command is addressed to a PPMID; Supplier otherwise

0x000F	CS04AC Method A or C Unjoin	`gSME`
		`eSME`
		`type1HANConnectedAuxiliaryLoadControlSwitch`
		`type1PrepaymentInterfaceDevice`
			Unjoin Device	SME.C.C	Access Control Broker where the Command is addressed to a PPMID; Supplier otherwise

0x0010	CS04B Method B Unjoin	`gSME` `eSME`	Unjoin Device	SME.C.NC	Supplier, Access Control Broker

[43] To avoid duplication of specification, the Use Cases here are grouped together, and the standard Use Case cross reference table is not used.

		communicationsHubGasProxyFunction			
0x0013	CS07 Read Device Join Details	gSME eSME communicationsHubGasProxyFunction type1HANConnectedAuxiliaryLoadControlSwitch type1PrepaymentInterfaceDevice		SME.C.N C	Supplier, Access Control Broker

Table 13.7.4.1: Join Device related Commands, Grouping and Message Categories

13.7.4.2 Join Device Command and Response Processing

13.7.4.2.1 Construction of Commands

'Join Device' Command Payloads shall be constructed as specified in Section 13.7.4.5.2 and Cryptographic Protection I and Cryptographic Protection II shall be applied as required for a Command of the relevant Message Category.

For a Command (1) which complies with either Use Case 'CS03A2 Method A Join (non Meter)' or Use Case 'CS03C Method C Join' and (2) where the Device to which it is addressed has a `deviceType` equal to `type1PrepaymentInterfaceDevice`, the Access Control Broker's Digital Signing Private Key shall be used in generating the KRP Signature.

13.7.4.2.2 Device processing of Commands and Response handling

The Device receiving a 'Join Device' Command shall undertake processing steps in the sequence defined in this Section 13.7.4.2.2. Should a step after step 1 be unsuccessful, the Device shall create a Response according to the requirements of Section 13.4.7, apply the Response Cryptographic Protection required for a Response of the relevant Message Category, and send the Response and shall not undertake any further steps defined in this Section 13.7.4.2.2.

In processing a 'Join Device' Command, the Device shall:

1. undertake Command Authenticity and Integrity Verification as required for a Command of this Message Category, except that check 4 in Section 6.2.4.1.1 may be undertaken after the checks in Section 6.2.4.1.2. The Security Credentials used to verify Cryptographic Protection I shall be:
   - those held in the {`accessControlBroker, digitalSignature, management`} Trust Anchor Cell, if `deviceType` equals `type1PrepaymentInterfaceDevice`; or
   - those held in the {`supplier, digitalSignature, management`} Trust Anchor Cell, if `deviceType` does not equal `type1PrepaymentInterfaceDevice`;
2. verify the `joinMethodAndRole` as specified in Section 13.7.4.5.3;
3. add the `otherDeviceEntityIdentifier` and `otherDeviceType` to its Device Log as specified in Section 13.7.4.5.4;
4. if `deviceType` is `eSME` then undertake Key Establishment with the other Device as specified in Section 13.7.4.5.5;
5. if `joinMethodAndRole` is `methodC`, and so the join is between a `gSME` and a `type1PrepaymentInterfaceDevice`, check that `otherDeviceCertificate` is present and validly structured. If the check succeeds the Device shall store, linked to this Device Log entry, details relating to `otherDeviceCertificate`, such that the Device is able to use subsequently the Shared Secret derived from `otherDeviceCertificate` and its own Private Key Agreement Key. If this check

fails the Device shall set `joinResponseCode` to `invalidOrMissingCertificate` and processing shall be unsuccessful; and

6. set `joinResponseCode` to `success`, create a Response according to the requirements of Section 13.4.7, apply the Response Cryptographic Protection required for a Response of the relevant Message Category, and send the Response.

13.7.4.2.3 Response Processing

Response Recipient Verification may be undertaken as specified in this GBCS for a Response of the relevant Message Category. The `joinResponseCode` field in the Response may be decoded according to the ASN.1 definitions at Section 13.7.4.5.1.

13.7.4.3 'Unjoin Device' Command and Response Processing

13.7.4.3.1 Construction of Commands

'Unjoin Device' Command Payloads shall be constructed as specified in Section 13.7.4.6.2 and Cryptographic Protection I and Cryptographic Protection II shall be applied as required for a Command of the relevant Message Category.

For a Command where the Device to which it is addressed has a `deviceType` equal to `type1PrepaymentInterfaceDevice`, the Access Control Broker's Digital Signing Private Key shall be used in generating the KRP Signature.

13.7.4.3.2 Device processing of Commands and Response handling

The Device receiving an 'Unjoin Device' Command shall undertake processing steps in the sequence defined in this Section 13.7.4.3.2.

In processing an 'Unjoin Device' Command, the Device shall:

1. undertake Command Authenticity and Integrity Verification as required for a Command of this Message Category. The Security Credentials used to verify Cryptographic Protection I shall be:
   - those held in the {`accessControlBroker, digitalSignature, management`} Trust Anchor Cell, if `deviceType` equals `type1PrepaymentInterfaceDevice`; or
   - those held in the {`supplier, digitalSignature, management`} Trust Anchor Cell, if `deviceType` does not equal `type1PrepaymentInterfaceDevice`;
2. set `unjoinResponseCode` to success;
3. verify the `otherDeviceEntityIdentifier` matches an Entity Identifier currently recorded in its Device Log. If it does not then set `unjoinResponseCode` to `otherDeviceNotInDeviceLog` and process from step 5; otherwise process from step 4;
4. delete all information from the entry in its Device Log that has the same Entity Identifier as `otherDeviceEntityIdentifier` along with all shared cryptographic material related to that entry. Additionally, If `deviceType` in that log entry is `type1HANConnectedAuxiliaryLoadControlSwitch`, the Device shall clear the association with the corresponding Auxiliary Controller [n], including updating the corresponding Auxiliary Controller [n] Type, with its SMETS meaning, to indicate that there is no associated Auxiliary Controller. If the deletion does not succeed, set `unjoinResponseCode` to `otherFailure`; and

5. Create a Response according to the requirements of Section 13.7.4.6.2, apply the Response Cryptographic Protection required for a Response of the relevant Message Category, and send the Response.

13.7.4.3.3 Response Processing

Response Recipient Verification may be undertaken as specified in this GBCS for a Response of the relevant Message Category. The `unjoinResponseCode` field in the Response may be decoded according to the ASN.1 definitions at Section 13.7.4.6.1.

13.7.4.4 'CS07 Read Device Join Details' Command and Response Processing

13.7.4.4.1 Construction of Commands

'CS07 Read Device Join Details' Command Payloads shall be constructed as specified in Section 13.7.4.7 and Cryptographic Protection II shall be applied as required for a Command of the SME.C.NC Message Category.

13.7.4.4.2 Device processing of Commands and Response handling

The Device receiving a 'CS07 Read Device Join Details' Command shall undertake processing steps in the sequence defined in this Section 13.7.4.4.2.

In processing a 'CS07 Read Device Join Details' Command, the Device shall:

1. undertake Command Authenticity and Integrity Verification as required for a Command of the SME.C.NC Message Category;
2. set `readLogResponseCode` to success;
3. attempt to read the Entity Identifier and `deviceType` for each of the entries in its Device Log. If the reading does not succeed for all entries, set `readLogResponseCode` to `readFailure`; otherwise populate `deviceLogEntries` using the data read from its Device Log; and
4. create a Response according to the requirements of Section 13.7.4.7, apply the Response Cryptographic Protection required for a Response of the SME.C.NC Message Category, and send the Response.

13.7.4.4.3 Response Processing

Response Recipient Verification may be undertaken as specified in this GBCS for a Response of the SME.C.NC Message Category. The `readLogResponseCode` and `deviceLogEntries` fields in the Response may be decoded according to the ASN.1 definitions at Section 13.7.4.7.

13.7.4.5 Component Requirements – Join

13.7.4.5.1 Join Command and Response payloads – structure definition

Each instance of `@JoinDevice.CommandPayload` and of `@JoinDevice.ResponsePayload` shall be an octet string containing the DER encoding of the populated structure defined in this Section 13.7.4.5.1 which specifies the structure in ASN.1 notation.

```
JoinDevice DEFINITIONS ::= BEGIN
CommandPayload ::=                                  SEQUENCE
{
     -- specify which type of joining is being authorised and,
     -- for Method A Joins, the role the Device is to play

     joinMethodAndRole                              JoinMethodAndRole,

     -- specify the Entity Identifier of the Device which is to be Joined with
```

```
     otherDeviceEntityIdentifier                    OCTET STRING,

     -- specify the DeviceType of that other Device

     otherDeviceType                                DeviceType,

     -- provide the other Device's Key Agreement certificate, if and only if this
     -- is a join between a gSME and a type1PrepaymentInterfaceDevice.
     -- Certificate shall be as defined in IETF RFC 5912

     otherDeviceCertificate                         Certificate OPTIONAL

}

-- detail whether the Command successful executed or, if it did not,
-- what the failure reason was

ResponsePayload ::=                                 JoinResponseCode

JoinMethodAndRole ::=                               INTEGER
{
     -- methodB is to be used where the other Device is a Type 2 Device or GPF.
     -- methodC is used where the Devices involved are a GSME and a PPMID.
     -- methodA is used otherwise.
     -- methodAInitiator is used where the Device this Command is targeted at
     -- should initiate the Key Agreement process
     -- methodAResponder is used where the Device this Command is targeted at
     -- should respond in the Key Agreement process, but shall not initiate it

     methodAInitiator                               (0),
     methodAResponder                               (1),
     methodB                                        (2),
     methodC                                        (3)
}

DeviceType ::=                                      INTEGER
{
     gSME                                           (0),
     eSME                                           (1),
     communicationsHubCommunicationsHubFunction     (2),
     communicationsHubGasProxyFunction              (3),
     type1HANConnectedAuxiliaryLoadControlSwitch    (4),
     type1PrepaymentInterfaceDevice                 (5),
     type2                                          (6)
}

JoinResponseCode::=                                 INTEGER
{
     success                                        (0),
     invalidMessageCodeForJoinMethodAndRole         (1),
     invalidJoinMethodAndRole                       (2),
     incompatibleWithExistingEntry                  (3),
     deviceLogFull                                  (4),
     writeFailure                                   (5),
     keyAgreementNoResources                        (6),
     keyAgreementUnknownIssuer                      (7),
     keyAgreementUnsupportedSuite                   (8),
     keyAgreementBadMessage                         (9),
     keyAgreementBadKeyConfirm                      (10),
     invalidOrMissingCertificate                    (11),
     noPartnerLinkKeyReceived                       (12),
     noCBKEResponse                                 (13)
}

END
```

13.7.4.5.2 Constructing the @JoinDevice.CommandPayload and of @JoinDevice.ResponsePayload

`@JoinDevice.CommandPayload` shall have the structure defined in Section 13.7.4.5.1, and the Remote Party constructing the Command 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`	`methodAInitiater (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.

Table 13.7.4.5.2a: `@JoinDevice.CommandPayload` population

`@JoinDevice.ResponsePayload` shall have the structure defined in Section 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	

Table 13.7.4.5.2b: `@JoinDevice.ResponsePayload` population

13.7.4.5.3 Verification of `joinMethodAndRole`

The Device shall first verify the `joinMethodAndRole` specified in the Command Payload against the Message Code specified in the Grouping 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 `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	CS03A2 Method A Join (non Meter)	`methodAResponder`
0x000E	CS03B Method B Join	`methodB`
0x00AF	CS03C Method C Join	`methodC`

Table 13.7.4.5.3a: Valid `deviceMethod` and `joinMethodAndRole` against Message Code

The Device receiving a Join Device Command shall verify `joinMethodAndRole` against its own `DeviceMethod` and the `DeviceType` specified in the `otherDeviceType` parameter of the Command according to the requirements of the remainder of this Section 13.7.4.5.3.

If `joinMethodAndRole` is `methodB` then the Device's verification of `joinMethodAndRole` shall be successful if there is a cell identified by 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 cell contains `'success'`. Otherwise, the verification shall fail and `JoinResponseCode` in the Response shall be set to the value `invalidJoinMethodAndRole`.

	otherDeviceType		
	communicationsHub GasProxyFunction	type1PrepaymentInterfaceDevice	type2
DeviceType of Device to which the Command is addressed			
gSME	Success	-	-
eSME	-	-	Success
communicationsHubGasProxyFunction	-	Success	Success

Table 13.7.4.5.3b: joinMethodAndRole is methodB

If joinMethodAndRole is methodAInitiator then the Device's verification of joinMethodAndRole shall be successful if there is a cell 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, and that cell contains 'success'. Otherwise, the verification shall fail and JoinResponseCode in the Response shall be set to the value invalidJoinMethodAndRole.

	otherDeviceType	
	type1HANConnected AuxiliaryLoadControlSwitch	type1Prepayment InterfaceDevice
DeviceType of Device to which the Command is addressed		
eSME	Success	Success

Table 13.7.4.5.3c: joinMethodAndRole is methodAInitiator

If joinMethodAndRole is methodAResponder then the Device's verification of joinMethodAndRole shall be successful if there is a cell 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, and that cell contains 'success'. Otherwise, the verification shall fail and JoinResponseCode in the Response shall be set to the value invalidJoinMethodAndRole.

	otherDeviceType
	eSME
DeviceType of Device to which the Command is addressed	
type1HANConnectedAuxiliaryLoadControlSwitch	Success
type1PrepaymentInterfaceDevice	Success

Table 13.7.4.5.3d: joinMethodAndRole is methodAResponder

If joinMethodAndRole is methodC then the Device's verification of joinMethodAndRole shall be successful if there is a cell identified by its own DeviceType and the value of otherDeviceType (as identified in the Command) in Table 13.7.4.5.3e and that cell contains 'success'. Otherwise, the verification shall fail and 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

Table 13.7.4.5.3e: joinMethodAndRole is methodC

13.7.4.5.4 Adding the otherDeviceEntityIdentifier and otherDeviceType to the Device Log

The Device shall undertake the following steps in the sequence specified:

1. if the otherDeviceEntityIdentifier matches an Entity Identifier currently recorded in its Device Log, then the Device shall compare deviceType in that log entry with otherDeviceType. If the Device types match then the addition is successful and processing within this Section 13.7.4.5.4 shall cease; otherwise the Device shall set joinResponseCode to incompatibleWithExistingEntry and processing within this Section 13.7.4.5.4 shall cease;
2. the Device shall check if there is capacity for an additional entry in its Device Log, including confirmation that, where otherDeviceType is type1HANConnectedAuxiliaryLoadControlSwitch the Device does not currently have an Auxiliary Controller associated with all

five of Auxiliary Controller [n]. If there is not, the Device shall set `joinResponseCode` to `deviceLogFull` and processing within this Section 13.7.4.5.4 shall cease; and

3. the Device shall attempt to create a new Device Log entry using `otherDeviceEntityIdentifier` and `otherDeviceType`. If that entry is not successfully created, the Device shall set `joinResponseCode` to `writeFailure`. If that entry is successfully created and `otherDeviceType` is `type1HANConnectedAuxiliaryLoadControlSwitch`, the Device shall associate one of its Auxiliary Controller [n] with this new HCALCS, including updating setting the corresponding Auxiliary Controller [n] Type, with its SMETS meaning, to indicate that this is an HCALCS whose Entity Identifier is `otherDeviceEntityIdentifier`.

13.7.4.5.5 Undertaking Key Establishment with the other Device

The Device shall undertake the following steps in the sequence specified:

1. if the `otherDeviceEntityIdentifier` does not match an Entity Identifier for which the ESME has received a Partner Link Key, the Device shall set `joinResponseCode` to `noPartnerLinkKeyReceived` and processing within this Section 13.7.4.5.5 shall cease; and

2. the ESME shall initiate, and attempt to complete, Key Establishment according to the ZSE requirements. The ESME shall wait a minimum of two seconds before timing out any key establishment operation. Should the first attempt to complete Key Establishment not succeed, the ESME shall wait 120 seconds before making a second attempt. If that second attempt fails, the ESME shall wait a further 120 seconds before making a final attempt. If that final attempt does not succeed, the Device shall set `joinResponseCode` to the value specified by Table 13.7.4.5.5 or to `noCBKEResponse` if no response was received from the other Device.

ZSE Response Code[44]	Value of `joinResponseCode`
NO_RESOURCES	`keyAgreementNoResources`
UNKNOWN_ISSUER	`keyAgreementUnknownIssuer`
UNSUPPORTED_SUITE	`keyAgreementUnsupportedSuite`
BAD_MESSAGE	`keyAgreementBadMessage`
BAD KEY_CONFIRM	`keyAgreementBadKeyConfirm`

Table 13.7.4.5.5: `joinResponseCode` mapping to ZSE Responses

[44] As defined in the ZSE specification

13.7.4.6 Component Requirements – Unjoin

13.7.4.6.1 Unjoin Command and Response payloads – structure definition

Each instance of `@UnjoinDevice.CommandPayload` and of `@UnjoinDevice.ResponsePayload` shall be an octet string containing the DER encoding of the populated structure defined in this Section 13.7.4.6.1 which specifies the structure in ASN.1 notation.

```
UnjoinDevice DEFINITIONS ::= BEGIN

CommandPayload ::=                          OtherDeviceEntityIdentifier
    -- specify the Entity Identifier of the Device for which authorisation
    -- is to be removed

    OtherDeviceEntityIdentifier ::=         OCTET STRING

ResponsePayload ::=                         UnjoinResponseCode

    -- detail whether the Command successful executed or, if it did not,
    -- what the failure reason was

UnjoinResponseCode::=                       INTEGER
{
   success                                  (0),
   otherDeviceNotInDeviceLog                (1),
   otherFailure                             (2)
}

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),`
		
		`(0),`
		`otherDeviceNotInDeviceLog (1),`
		
		`(1),`
		`otherFailure (2)`
		
		`(2)`
			Mandatory	Shall be populated according to the processing defined in Section 13.7.4.3

Table 13.7.4.6.2b: `@UnjoinDevice.ResponsePayload` population

13.7.4.7 CS07 Read Device Join Details Command and Response payloads – structure definition

Each instance of `@ReadDeviceLog.CommandPayload` and of `@ReadDeviceLog.ResponsePayload` shall be an octet string containing the DER encoding of the populated structure defined in this Section 13.7.4.7 which specifies the structure in ASN.1 notation.

```
ReadDeviceLog DEFINITIONS ::= BEGIN

CommandPayload ::=                              NULL

ResponsePayload ::=                             SEQUENCE
{
    -- detail whether the Command successful

    readLogResponseCode                         ReadLogResponseCode,

    -- if it was, return the Log Entries
    deviceLogEntries                            SEQUENCE OF DeviceLogEntry OPTIONAL
}

DeviceLogEntry ::=                              SEQUENCE
{
    deviceIndentifier                           OCTET STRING,
    deviceType                                  DeviceType
}

DeviceType ::=                                  INTEGER
{
   gSME                                         (0),
   eSME                                         (1),
   communicationsHubCommunicationsHubFunction   (2),
   communicationsHubGasProxyFunction            (3),
   type1HANConnectedAuxiliaryLoadControlSwitch  (4),
   type1PrepaymentInterfaceDevice               (5),
   type2                                        (6)
}


ReadLogResponseCode::=                          INTEGER
{
   success                                      (0),
   readFailure                                  (1)
}

END
```

13.8 GCS59 / 62 GPF Device Log Backup and Restore

13.8.1 Introduction to GPF Device Log Backup and Restore – informative

13.8.1.1 The role of pair-wise authorisation – informative

This Section 13.8 includes the Use Cases related to the backing up and restoring of the GPF's Device Log. This is to cater for situation where the existing Communications Hub fails and has to be replaced.

In summary:

- a GPF sends an Alert whenever its Device Log changes (unless the change is as a result of a restore of the Device Log). That Alert contains the contents of the GPF's Device Log after the change has been made; and
- the Restore GPF Device Log Command shall contain the same structure of Device Log contents. If successful, the Command will place those contents in to the GPF's Device Log and will have triggered the processing required to authorise the specified Devices application layer interaction with the GPF, where required.

13.8.1.2 The format of Message Payloads – informative

In common with other GBCS Remote Messages related to the management of Security Credentials, the Payloads of Alerts, Commands and Responses defined in this Section 13.8 are specified using ASN.1, with DER encoding to be applied to Command and Response payloads.

Each entry in a GPF Device Log shall contain the Entity Identifier of the Authorised Device and its `deviceType`.

13.8.2 GCS62 Backup GPF Device Log

13.8.2.1 Description

This Section 13.8.2 covers the creation, validation and processing of Alerts resulting from changes to the GPF Device Log. One such Alert shall be generated each time that the GPF Device Log changes, except where the change arises from a GPF Device Log Restore Command.

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
Service 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	N/A

Cross Reference	Value
Broker on behalf of parties not known to the Device) [Remote Party Messages Only]	
Valid initiating Device type(s) [HAN Only Messages]	N/A
Protocol	ASN.1

Table 13.8.2.2: Use Case Cross References for GPF Device Log Backup Alert

13.8.2.3 Construction of Alerts

GPF Device Log Backup Alert Payloads shall be constructed according to the requirements of Section 13.8.4.1 and populated as specified in Table 13.8.2.3.

MAC Header, Grouping Header and SMD-KRP MAC shall be populated as required for an Alert of the SME.A.NC Message Category, with the Message Code being 0x00B2. Note that the Business Target ID in the Grouping Header shall always contain the Entity Identifier of 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`	0x8071	Mandatory	Fixed value specifying that this is a GPF Device Log Backup Alert
`backupDateTime`	`Generalized Time`	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 6.8.4.

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
Service 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

Table 13.8.3.2: Use Case Cross References for GPF Device Log Restore

13.8.3.3 Construction of Command

GPF Device Log Restore Command Payloads shall be constructed according to the requirements of Section 13.8.4.1 and populated as specified in Table 13.8.3.3.

MAC Header, Grouping Header and ACB-SMD MAC shall be populated as required for a Command of the SME.C.NC 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
`@GPFDeviceLog.RestoreCommandPayload`	`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.

Table 13.8.3.3: @GPFDeviceLog.RestoreCommandPayload population

13.8.3.4 Device processing of Command and Response handling

The GPF receiving a GPF Device Log Restore Command shall undertake processing steps in the sequence defined in this Section 13.8.3.4. The Device shall undertake Command Authenticity and Integrity Verification as required for a Command of this Message Category, and then, if successful, for each DeviceLogEntry in deviceLogEntries, shall:

1. set deviceLogEntry in the corresponding ResponseOutcome to the values of this DeviceLogEntry in deviceLogEntries;
2. set joinResponseCode in the corresponding ResponseOutcome to success;
3. 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;
4. 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
5. 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.

Once all DeviceLogEntry in deviceLogEntries have been processed, the GPF shall populate the Response Payload according to the requirements of Section 13.8.4.1 using the ResponseOutcomes produced by the processing in this Section 13.8.3.4, construct MAC Header, Grouping Header and apply the Response Cryptographic Protection required for a Response of the SME.C.NC Message Category, and send the Response.

13.8.4 Common Requirements

13.8.4.1 GPF Device Log Backup Alert, Restore Command and Restore Response Payloads – structure definition

Each instance of `@GPFDeviceLog.BackupAlertPayload`, `@GPFDeviceLog.RestoreCommandPayload` and of `@GPFDeviceLog.RestoreResponsePayload` shall be an octet string containing the DER encoding of the populated structure defined in this Section 13.8.4.1 which specifies the structure in ASN.1 notation.

```
GPFDeviceLog DEFINITIONS ::= BEGIN

BackupAlertPayload ::=                              SEQUENCE
{
     -- specify the Alert Code
     alertCode                                      INTEGER(0..4294967295),

     -- specify the date-time of the backup
     backupDateTime                                 GeneralizedTime,

     -- detail the entries in the Device Log now that the change has been made
     deviceLogEntries                               SEQUENCE OF DeviceLogEntry

}

RestoreCommandPayload ::=                           SEQUENCE
{
     -- list the Device Log entries that are to be added
     deviceLogEntries                               SEQUENCE OF DeviceLogEntry

}

DeviceLogEntry ::=                                  SEQUENCE
{

     -- specify the Entity Identifier of the Device
     deviceEntityIdentifier                         OCTET STRING,

     -- specify the DeviceType of that Device

     deviceType                                     DeviceType
}

RestoreResponsePayload ::=                          SEQUENCE
```

```
{
     -- for each DeviceLog Entry, detail whether the Command successfully executed or, if it did not, what the failure reason
was

     restoreOutcomes                                SEQUENCE OF RestoreOutcome
}

RestoreOutcome ::=                                  SEQUENCE
{
     deviceLogEntry                                 DeviceLogEntry,
     joinResponseCode                               JoinResponseCode
}

DeviceType ::=                                      INTEGER
{
     gSME                                           (0),
     eSME                                           (1),
     communicationsHubCommunicationsHubFunction     (2),
     communicationsHubGasProxyFunction              (3),
     type1HANConnectedAuxiliaryLoadControlSwitch    (4),
     type1PrepaymentInterfaceDevice                 (5),
     type2                                          (6)
}

JoinResponseCode::=                                 INTEGER
{
     success                                        (0),
     invalidMessageCodeForJoinMethodAndRole         (1),
     invalidJoinMethodAndRole                       (2),
     incompatibleWithExistingEntry                  (3),
     deviceLogFull                                  (4),
     writeFailure                                   (5),
     keyAgreementNoResources                        (6),
     keyAgreementUnknownIssuer                      (7),
     keyAgreementUnsupportedSuite                   (8),
     keyAgreementBadMessage                         (9),
     keyAgreementBadKeyConfirm                      (10),
     invalidOrMissingCertificate                    (11)
}
END
```

13.9 CS02f Provide Security Credential Details Command and Response

13.9.1 Description

This Section covers the creation, validation and processing of (i) CS02f Provide Security Credential Details Commands and (ii) Responses to such Commands.

13.9.2 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
Service Reference	6.24.1
Valid Target Device(s)	ESME / SAPC
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 Load Controller
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

Table13.9.2: Use Case Cross References for CS02f Provide Security Credential Details Command and Response

13.9.3 Common Requirements

13.9.3.1 Summary – informative

The statements in Section 13.2.3.1 apply.

13.9.3.2 The 'CS02f Provide Security Credential Details' Command and Response

The requirements of Section 13.2.3.2 shall apply as if `@ProvideSecurityCredentialDetails` were modified as required by Section 13.2.3.2, and references to sections within Section 13.2 were references to the corresponding sections within Section 13.9.

13.9.3.3 The @ProvideSecurityCredentialDetails.Command and @ProvideSecurityCredentialDetails.Response structure definition

The requirements of Section 13.2.3.3 shall apply as if:

- The reference to Section 13.2.4.4 were a reference to Section 13.9.4.4; and
- `RemotePartyRole` were defined as:

```
RemotePartyRole ::=              INTEGER
{

root                             (0),
recovery                         (1),
supplier                         (2),
networkOperator                  (3),
accessControlBroker              (4),
transitionalCoS                  (5),
wanProvider                      (6),
issuingAuthority                 (7),
loadController                   (8),
other                            (127)

}
```

13.9.4 CS02f Provide Security Credential Details from a Device – Processing Steps

The requirements of Section 13.2.4, and all sub-sections of it, shall apply as if:

- 'CS02a Provide Security Credential Details' were 'CS02f Provide Security Credential Details'';
- `RemotePartyRole` were defined as required by Section 13.9.3.3;
- references to Section 13.2.3.3 were references to Section 13.9.3.3; and
- the allowed values for `trustAnchorCellRemotePartyRole` and `RemotePartyRole` in the Command, and `remotePartyRole` in the Response included `loadController (8)`.

13.9.4.1 Command Construction

See Section 13.9.4.

13.9.4.2 Command Cryptographic Protection

See Section 13.9.4.

13.9.4.2.1 Access Control Broker MAC

See Section 13.9.4.

13.9.4.3 Command Authenticity and Integrity Verification

See Section 13.9.4.

Additionally, the requirements shall apply as if the Message Code required in 'Check Number 1.2' of Table 13.2.4.3 were 0x011B.

13.9.4.3.1 Command Authenticity and Integrity Verification

See Section 13.9.4.

13.9.4.4 Response Construction

See Section 13.9.4.

13.9.4.5 Response Cryptographic Protection

See Section 13.9.4.

13.9.4.6 Response Recipient Cryptographic Verification

See Section 13.9.4.

13.10 CS02g Update Load Controller Security Credentials Command, Response and Alert

13.10.1 Description

See Section 13.3.1.

13.10.2 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
Protection Against Replay Required?	The Protection Against Replay mechanisms for Update Security Credentials are specified in Section 13.10. The Protection Against Replay mechanisms of other Sections of the GBCS do not apply
Service Reference	6.15.1
Valid Target Device(s)	ESME / SAPC
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 LoadController
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

Table 13.3.2: Use Case Cross References for Update Security Credentials Command

13.10.3 Command, Response and Alert Structure

13.10.3.1 The Update Load Controller Security Credentials Command

The Update Load Controller Security Credentials Command shall be the concatenation:

MAC Header || Grouping Header || `@UpdateLoadControllerSecurityCredentials.CommandPayload` || 0x40 || KRP Signature|| ACB-SMD MAC

with the requirements of Section 13.3.3.1, as to the components of that concatenation, applying as if `@UpdateLoadControllerSecurityCredentials` were `@UpdateSecurityCredentials`.

13.10.3.2 The Update Load Controller Security Credentials Response

An Update Security Credentials Response shall be the concatenation:

Grouping Header || @UpdateLoadControllerSecurityCredentials.ResponsePayload || 0x40 || SMD Signature

with the requirements of Section 13.3.3.2, as to the components of that concatenation, applying as if @UpdateLoadControllerSecurityCredentials were @UpdateSecurityCredentials.

13.10.3.3 Future Dated Update Load Controller Security Credentials Alert

A Future Dated Update Load Controller Security Credentials Alert shall be the concatenation:

Grouping Header || @UpdateLoadControllerSecurityCredentials.AlertPayload || 0x40 || SMD Signature

where:

- @UpdateLoadControllerSecurityCredentials.AlertPayload shall be an octet string containing the DER encoding of the populated ASN.1 structure;
- Grouping Header in the Alert shall be constructed as specified in Section 7.2.7 with Business Target ID being the Entity Identifier specified in the corresponding Command's Grouping Header;
- the Message Code being 0x0124; and
- SMD Signature shall be generated as specified in Section 6.3.5.

13.10.3.4 The Update Load Controller Security Credentials Command, Response and Alert – informative

The statements of Section 13.3.3.4 shall apply as if:

- @UpdateLoadControllerSecurityCredentials were @UpdateSecurityCredentials; and
- the reference to Section 13.3.5.1 were a reference to Section 13.10.5.1

save for the fact that Section 18.3.1 does not provide an an illustrative instantiation of @UpdateLoadControllerSecurityCredentials.CommandPayload.

13.10.4 Updating Load Controller Security Credentials on a Device – Processing Steps

This Section lays out the requirements for the construction, protection and Authentication of the Update Load Controller Security Credentials Command Payload, the processing required on the Device of the Command, the construction of the corresponding Response Payload and, where required, the Alert Payload.

13.10.4.1 Command Payload construction

The @ UpdateLoadControllerSecurityCredentials.CommandPayload shall have the structure defined in Section 13.10.5.11, and the Remote Party constructing the Command shall populate with values according to Table 13.3.4.1 as if:

- @UpdateLoadControllerSecurityCredentials were @UpdateSecurityCredentials;

- The allowed values for `credentialsReplacementMode` were either `loadControllerByLoadController (10)` or `loadControllerBySupplier (11)`;
- Within `authorisingRemotePartyTACellIdentifier` the allowed values for trustAnchorCellRemotePartyRole were either `supplier (2)` or `loadController (8)`;
- The `otherRemotePartySeqNumberChanges` sequence's presence were Mandatory in all `loadControllerBySupplier (11)` Commands, since all such Commands effect a change of control;
- When present, the only allowed value for `otherRemotePartyRole` were `loadController (8)`;
- Within `targetTrustAnchorCell`, the allowed value for `trustAnchorCellRemotePartyRole` were `loadController (8)`; and
- `executionDateTime` were allowed to be present when `credentialsReplacementMode` were either `loadControllerByLoadController (10)` or `loadControllerBySupplier (11)`.

13.10.4.2 Command Authenticity and Integrity Verification

The Device shall undertake processing according to the requirements of Section 13.10.5.1.

Should any of the checks detailed in Section 13.10.5.1 fail then the Device shall:

- generate an entry in the Security Log recording failed Authentication;
- discard the Command without execution and without sending a Response; and
- 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 the Trust Anchor Cell {`supplier, digitalSignature, management`}.

Where all of the checks detailed in Section 13.10.5.1 succeed the Device shall process the Command and produce a Response.

13.10.4.3 Command Processing

Before undertaking any further processing, the Device shall update the Execution Counter for this type of Command to the value of `authorisingRemotePartySeqNumber`.

If `executionDateTime` is present then the Device shall:

- record against the `remotePartyRole` (as specified in `authorisingRemotePartyControl`), `authorisingRemotePartyControl`, `replacements`; and `executionDateTime`;
- construct a Response where `executionOutcome` is not present according to the requirements of Section 13.10.4.4; and
- at the date-time specified in `executionDateTime`, or immediately if the `executionDateTime` is in the past undertake the processing of Section 13.10.4.3.1 then construct an Alert according to the requirements of Section 13.10.4.5.

If `executionDateTime` is not present then the Device shall:

- undertake the processing of Section 13.10.4.3.1; and
- construct a Response where `executionOutcome` is present according to the requirements of Section 13.10.4.4.

13.10.4.3.1 `replacements` Processing

For each of the `targetTrustAnchorCell` in `replacements`, the Device shall:

- record the `entityIdentifier` and `subjectKeyIdentifier` currently held in that `targetTrustAnchorCell`;
- attempt to replace the contents of that `targetTrustAnchorCell` using the corresponding `certificate` in `TrustAnchorReplacement`; and
- if the contents of the `replacement` are successfully applied, undertake the processing required by Section 13.10.5.10 in relation to the `RemotePartyRole` for that `targetTrustAnchorCell`.

For clarity, before the contents of a certificate are used to replace the contents of a `targetTrustAnchorCell`, the Device shall have successfully undertaken the checks in Section 13.10.5.9 in relation to that certificate. Should a check in 13.10.5.9 not succeed then, in line with Section 4.3.2.8.2, the Device shall undertake no further processing of the Command, except for the issuance of a Message containing an `executionOutcome`.

13.10.4.4 Response Construction

The `@UpdateLoadControllerSecurityCredentials.ResponsePayload` shall have the structure defined in Section 13.10.5.11, and the Device shall populate the `executionOutcome`, where present with values according to Section 13.10.4.6.

13.10.4.5 Alert Construction

The `@UpdateLoadControllerSecurityCredentials.AlertPayload` shall have the structure defined in Section 13.10.5.11, and the Device shall populate the `executionOutcome`, with values according to Table 13.3.4.6.

13.10.4.6 `executionOutcome` construction

The requirements of Section 13.3.4.6 shall apply as if:

- The allowed values for `credentialsReplacementMode` were either `loadControllerByLoadController (10)` or `loadControllerBySupplier (11)`;
- When present, the only allowed value for `otherRemotePartyRole` were `loadController (8)`; and
- Within `affectedTrustAnchorCell`, the allowed value for `trustAnchorCellRemotePartyRole` were `loadController (8)`.

13.10.5 Common Requirements

13.10.5.1 Update Security Credentials Command Verification

The Device shall undertake the checks set out in this Section 13.10.5.1 before undertaking any other processing of the Command. The checks may be carried out in any order. Checking shall cease at the point that any one check fails. The checks required are shown in Table 13.10.5.1.

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 13.10.5.2 for the `CredentialsReplacementMode` specified in `CommandPayload`
1.4	The Device has not already actioned this Command.	As specified in Section 13.10.5.3
2.1	The `targetTrustAnchorCells` all exist on a Device of this type	As specified in Section 13.10.5.4
2.1	The `trustAnchorCellUsage` in the `authorisingRemotePartyTACellIdentifier` must be ABSENT	As specified in Section 13.10.4.1
2.2	The `credentialsReplacementMode` is one that can be Authorised by the Remote Party / Parties authorising the Command	As specified in Section 13.10.5.5
2.2	The `replacements` specified are all allowed in this `credentialsReplacementMode`.	As specified in Section 13.10.5.6
2.2	The `TrustAnchorCellUsage` in the `targetTrustAnchorCell` in each entry in replacements must be ABSENT unless it has the value `prePaymentTopUp` as specified in the Notes column of Table 13.3.4.1	As specified in Section 13.10.4.1
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 13.10.5.7
3.1	The Cryptographic Protections are valid	As specified in Section 13.10.5.8

Table 13.10.5.1: Update Load Control Security Credentials Command authenticity and integrity verification

13.10.5.2 Message Code Validation

`CredentialsReplacementMode`	Message Code
`loadControllerByLoadController`	0x0125
`loadControllerBySupplier`	0x0126

Table 13.10.5.2: Message Code validation against `CredentialsReplacementMode`

13.10.5.3 Preventing Replay of Commands

The Protection Against Replay mechanisms for the Update Security Credentials Command shall be that specified in this Section 13.10.5.3 (which is different than that for other GBCS Commands).

For each of RemotePartyRole from which the Device can receive a valid Updated Load Controller Security Credentials Command, the Device shall allocate storage for an Execution Counter which shall be capable of storing a 64 bit unsigned integer and which shall initially be set to the value zero at manufacture.

Before executing any Update Security Credentials Command, a Device shall confirm that (authorisingRemotePartyTACellIdentifier is populated in the Command) and (the authorisingRemotePartySeqNumber is strictly numerically greater than the Execution Counter the Device has recorded for the RemotePartyRole identified in authorisingRemotePartyTACellIdentifier).

13.10.5.4 Required Trust Anchor Cells by Device Type

The Trust Anchor Cells specified in Section 4.3.2.5 for loadController are those required on each Device type and so are the only valid targetTrustAnchorCells.

The Device shall ensure that all targetTrustAnchorCells specified in the Command instance are valid for the type of Device it is, according to the requirements of Section 4.3.2.5. A Command containing any invalid targetTrustAnchorCells shall not be processed any further by the Device.

13.10.5.5 Valid credentialsReplacementMode by Remote Party Roles authorising the Command

A Command containing a certain credentialsReplacementMode is only Authorised using certain types of Public-Private Key Pairs in certain ways. The Command identifies the Public Key corresponding to the Private Key used by the authorising Remote Party in the authorisingRemotePartyTACellIdentifier structure. Table 13.10.5.5 lists the only Authorised combinations. All other combinations represent Commands not properly Authorised and shall be rejected by a Device.

	authorisingRemotePartyTACellIdentifier		
	RemotePartyRole	KeyUsage	CellUsage
credentialsReplacementMode			
loadControllerBySupplier	supplier	digitalSignature	management
loadControllerByLoadController	loadController	digitalSignature	management

Table 13.10.5.5: Valid credentialsReplacementMode by Remote Party Roles authorising the Command

13.10.5.6 Valid credentialsReplacementMode by the targetTrustAnchorCells specified in the Command

A Command containing a certain credentialsReplacementMode can only validly replace the Security Credentials in a certain subset of Trust Anchor Cells. The Command identifies the Cells that are to have credentials replaced in each targetTrustAnchorCell within each TrustAnchorReplacement in replacements.

Table 13.10.5.6 below lists the only valid targetTrustAnchorCell combinations for each credentialsReplacementMode. All other combinations are invalid. A Command containing any invalid combinations shall not be processed any further by the Device.

	targetTrustAnchorCell		
	RemotePartyRole	KeyUsage	CellUsage
credentialsReplacementMode			
loadControllerBySupplier	loadController	any valid for GBCS	any valid for GBCS
loadControllerByLoadController	loadController	any valid for GBCS	any valid for GBCS

Table 13.10.5.6: Valid `credentialsReplacementMode` by the `targetTrustAnchorCells` specified in the Command

13.10.5.7 Valid usage of Certificates against the `targetTrustAnchorCells` specified in the Command

For each instance of the `TrustAnchorReplacement` structure in the Command, the `keyUsage` in `replacementCertificate` shall be equal to `targetTrustAnchorCell.KeyUsage`. Where this check fails for any one or more of the `TrustAnchorReplacement` instances, the Command shall not be actioned by the Device.

Note that mismatches between `RemotePartyRole` in the certificate and the target Trust Anchor Cell are admissible for `loadController`, and are needed (see Section 4.3.2.5).

13.10.5.8 Verifying the Cryptographic Protections

In verifying Cryptographic Protections:

- KRP Signature shall be verified using the public key identified in the Command; and
- ACB-SMD MAC shall be verified according to the requirements in Section 6.2.4.1.2.

13.10.5.9 Verifying the authenticity of replacement certificates

The Device shall first apply the requirements of Section 12.6 (Device processing of Certificates). If any of those checks fail, the Section 13.10.5.9 check fails.

In relation to Certification Path Validation, the application of time based checks shall be determined as follows:

- if, in the Command, `applyTimeBasedCPVChecks` = `disapply` then time based checks shall NOT be applied by the Device; and
- otherwise time based checks shall be applied or not applied in line with the requirements of Section 4.3.2.8.

The Device shall, for each `replacementCertificate` in `replacements`, undertake Certification Path Validation according to the requirements of Section 4.3.2.8.

13.10.5.10 Required Processing on Change of Remote Party Control

If the `targetTrustAnchorCell` is `{loadController, digitalSignature, management}` and there are any future dated Update Load Controller Security Credentials held on the Device that have not yet executed, and so their `executionDateTime` is in the future, then the Device shall set each such `executionDateTime` to '99991231235959Z'.

If:

- the `targetTrustAnchorCell` is `{loadController, digitalSignature, management};` and
- the Entity Identifier in the `targetTrustAnchorCell` is changed by the replacement

then the Device shall set the execution date-time of any other future dated Commands that are from the Load Controller and that are held on the Device but not yet executed, to 'never', as detailed in Section 9.2.

Relevant Execution Counters' shall be those which relate to types of Commands:

- which originate from the `remotePartyRole` of this `targetTrustAnchorCell`;
- whose cryptographic protection is checked using the Public Key in this `targetTrustAnchorCell`.

Where this Command contains an `otherRemotePartySeqNumberChange` in which `otherRemotePartyRole` is the `remotePartyRole` of this `targetTrustAnchorCell`, the Device shall set all Relevant Execution Counters to be the corresponding value in `otherRemotePartyFloorSeqNumber`.

If:

- `remotePartyRole` of `targetTrustAnchorCell` and that of `authorisingRemotePartyControl` is `loadController`; and
- `keyUsage` of `targetTrustAnchorCell` is `digitalSignature`,

then the Device shall set all Relevant Execution Counters to the value in `newRemotePartyFloorSeqNumber`.

Note, where `SeqNumber` values are to be used to populate Execution Counters on the Device, these `SeqNumber` values shall be processed as unsigned 64 bit integers, so 8 octets in length.

13.10.5.11 The `@UpdateLoadControllerSecurityCredentials.CommandPayload`, `@UpdateLoadControllerSecurityCredentials.ResponsePayload` and `@UpdateLoadControllerSecurityCredentials.AlertPayload` structure definition

The requirements of Section 13.3.5.11 shall apply as if:

- `@UpdateLoadControllerSecurityCredentials` were `@UpdateSecurityCredentials`;
- `RemotePartyRole` were defined as:

```
RemotePartyRole ::=              INTEGER
{

root                             (0),
recovery                         (1),
supplier                         (2),
networkOperator                  (3),
accessControlBroker              (4),
transitionalCoS                  (5),
wanProvider                      (6),
issuingAuthority                 (7),
loadController                   (8),
other                            (127)

}
```

- and `CredentialsReplacementMode` were defined as:

```
CredentialsReplacementMode ::=   INTEGER
```

```
{
supplierBySupplier                (2),
networkOperatorByNetworkOperator (3),
accessControlBrokerByACB          (4),
wanProviderByWanProvider          (5),
transCoSByTransCoS                (6),
supplierByTransCoS                (7),
anyExceptAbnormalRootByRecovery   (8),
anyByContingency                  (9),
loadControllerByLoadController    (10),
loadControllerBySupplier          (11)
}
```

14 Apply Prepayment Top Up to an ESME or GSME

Requirements in this Section 14, and sub-Sections of it, shall, where the SAPC supports the corresponding SMETS functionality, apply to an SAPC as if it were an ESME.

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 Prepayment 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 20th digit of the UTRN
UTRN Counter Cache	Shall be an array capable of containing up to 100 entries, each entry containing an unsigned integer of 32 bits in length. At manufacture, the UTRN Counter Cache shall contain at least one zero value and shall contain no values which are non-zero. 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

- applying a Prepayment Top Up to an ESME or GSME with consumer entry of a numeric code on a PPMID.

Some requirements are common to all four options. Accordingly, this Section 14 is split in to five subsections:

- an initial subsection covering requirements common to all four options; and
- four subsequent subsections covering one option in each subsection.

By way of context:

- any Prepayment Top Up Message is a Remote Party Command in GBCS terms (because it is from a Remote Party to a GSME or ESME). The means of delivery (typing in on meter, typing in on PPMID, sending over WAN, etc.) does not affect this classification;
- as a Remote Party Command, it must result in the GSME or ESME generating a Response back to the Remote Party who issued it (so the Supplier), unless there is an Authentication failure (in which case the Supplier has to be sent an Alert), as per the Technical Specifications;
- because the ranges are exclusive, the Originator Counter in Prepayment Top Up transactions cannot collide with the Originator Counter in any other transaction; and
- there is no requirement to include the Device's ID explicitly in the locally entered transaction, so a PPMID joined to more than one Smart Meter will need to allow the Consumer to pick which Smart Meter the Prepayment Top Up is for.

14.3 Common Requirements

14.3.1 Construction of the PPTD

The PPTD shall be a 19 decimal digit integer. The most significant two digits of the PPTD shall always be between 73 and 96, which shall be constructed and represented according to the requirements of this Section 14.3.1.

The decimal representation of the PPTD shall be the result of the addition of 7,394,156,990,786,306,048 to the decimal representation of the PTUT.

14.3.2 Construction of the PTUT

The PTUT shall be an unsigned 64 bit integer (so 8 octets), which shall be constructed and represented according to the requirements of this Section 14.3.2.

The bits within the PTUT shall be numbered from 63 for the most significant bit through to 0 for the least significant bit.

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 Truncated Originator Counter	Bits 41-32 of the Originator Counter	56-47	Used for Protection Against Replay purposes when the transaction is entered locally
PTUT Value Class	0b00 if PTUT Value is to be interpreted as multiples of 1/100 of Currency Unit; OR	46-45	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 component	Value	Bits	Note
	0b01 if PTUT Value is to be interpreted as multiples of Currency Unit.		PTUT Value will be interpreted as GB Pounds
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	44-32	Thus, the maximum value is either: £81.91 if PTUT Value Class =0b00; or £8,191.00 if PTUT Value Class =0b01.
PTUT Supplier MAC	See Section 14.3.4	31-0	

Table 14.3.2: Values of PTUT bits

14.3.3 Construction of the Unique Transaction Reference Number (UTRN)

The Unique Transaction Reference Number (UTRN) shall be a 20 decimal digit which shall be the 19 decimal digits of the PPTD with a 20th decimal digit which shall be appended after the least significant digit of the 19 decimal digit representation of PPTD. This 20th decimal digit shall be the UTRN Check Digit. The UTRN Check Digit shall be calculated according to the requirements of Section 14.8.

14.3.4 Construction of the PTUT Supplier MAC

The PTUT Supplier MAC shall only be calculated once the 32 most significant bits of PTUT (bits 63-32 of the PTUT) have been populated as per the requirements of Section 14.3.2.

The Remote Party, whose Security Credentials are stored against the Supplier role of the target Device, shall calculate a MAC using the parameters in Table 14.3.4 then setting the PTUT Supplier MAC to be the 32 least significant bits of the 128 bit MAC produced by the 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:	0x110000000000 || Supplier's `prePaymentTopUp` Entity Identifier || Device's Entity Identifier || 0x01 || Originator Counter || 32 most significant bits of the PTUT	

Table 14.3.4: Calculation of the PTUT Supplier MAC

14.3.5 Validating the PTUT Supplier MAC

To validate the PTUT Supplier MAC, the Device shall calculate the MAC using the parameters in Table 14.3.5, then ensure the 32 least significant bits of the 128 bit MAC produced by the MAC calculation has the same value as the PTUT Supplier MAC.

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:	0x110000000000 || Supplier's `prePaymentTopUp` Entity Identifier || Device's Entity Identifier || 0x01 || Originator Counter || 32 most significant bits of the PTUT	

Table 14.3.5: Validation of the PTUT Supplier MAC

14.3.6 Checking the UTRN Counter against the UTRN Counter Cache

The Device shall set the UTRN Counter to be the 32 most significant bits of the Originator Counter.

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 Counter Cache.

14.3.7 Updating the UTRN Counter Cache

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.

14.3.8 Validating the Maximum Credit Values

14.3.8.1 Maximum Credit Threshold

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.

14.3.8.2 Maximum Meter Balance Threshold

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.

14.3.9 Validating the PTUT Sub-Class

The Device shall ensure that the value specified by PTUT Sub-Class is of value 0b0000.

14.3.10 Applying Prepayment Top Ups

For clarity, ESME and GSME shall apply the value of Prepayment Top Ups as specified in SMETS.

14.4 CS01a Applying a Prepayment Top Up to an ESME without consumer intervention

14.4.1 Description

This Section covers the application of a Prepayment Top Up that has been bought for an ESME to that ESME, in the case where the consumer does not enter any details on Devices in their premises.

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
Service 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

14.4.3 Pre-conditions

None.

14.4.4 Detailed Steps

The Device shall undertake the checks set out in this Section 14.4.4 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.

If any of the checks specified in this Section 14.4.4 fail, the Device shall not carry out further checks, and the requirements of Section 6.2.4.2 shall apply. Otherwise, processing shall continue as per the requirements of Section 6.2.4.1.2. Where that check is successful, processing shall continue as below.

14.4.4.1 Verifying against the maximum credit values

The Device shall carry out the checks specified in Section 14.3.8.1 and Section 14.3.8.2.

14.4.4.2 Verifying the Originator Counter

The Device shall verify the Originator Counter against the UTRN Counter Cache according to Section 14.3.6.

14.4.4.3 Validating the PTUT Supplier MAC

The Device shall validate the PTUT Supplier MAC according to Section 14.3.5.

14.4.5 Response Construction

At the Response Construction stage, the Device shall first update the UTRN Counter Cache according to Section 14.3.7, and shall then populate the Response according to the requirements of the Message Template for CS01a.

14.5 CS01b Applying a Prepayment Top Up to a GSME without consumer intervention

14.5.1 Description

This Section covers the application of a Prepayment Top Up that has been bought for a GSME to that GSME, in the case where the consumer does not enter any details on Devices in their premises, except for the additional processing defined in this Section.

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
Service 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
Protocol	GBZ

Table 14.5.2: Use Case Cross References for Prepayment Top Up to a GSME without consumer intervention

14.5.3 Pre-conditions

None.

14.5.4 Detailed Steps

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

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

The Device shall verify the Originator Counter against the UTRN Counter Cache according to Section 14.3.6.

14.5.4.3 Validating the PTUT Supplier MAC

The Device shall validate the PTUT Supplier MAC according to Section 14.3.5.

If any of the checks specified in this Section 14.5.4 fail, the requirements of Section 6.2.4.2 shall apply. Otherwise, processing shall continue as per the requirements of Section 6.2.4.1.2.

14.5.5 Response Construction

At the Response Construction stage, the Device shall first update the UTRN Counter Cache according to Section 14.3.7 and shall then populate the Response according to the 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

This Section covers the application of a Prepayment Top Up that has been bought for a GSME or ESME to that GSME or ESME in the case where the consumer enters the corresponding UTRN on the GSME or ESME.

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

Cross Reference	Value
	with Section 14.3.3. The Command includes cryptographic protections as specified in Sections 14.3.4 and 14.3.5. The Response shall be of Message Type Category SME.C.NC.
Capable of future dated invocation?	No
Protection Against Replay Required?	See Table 14.4.2
Service 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

14.6.3 Pre-conditions

None.

14.6.4 Detailed Steps

14.6.4.1 Detailed Steps/Sequence

The Device shall undertake the validation checks set out in this Section 14.6.4.1 before undertaking any other processing of the Command. The validation checks shall be undertaken in the sequence laid out. Should a validation check fail, subsequent validation checks shall not be undertaken by the Device.

Should any of the checks fail (save for the optional UTRN Check Digit verification), the requirements of Section 6.2.4.2 shall apply.

14.6.4.1.1 Verifying the UTRN Check Digit

The Device:

- may validate the 20th digit (the UTRN Check Digit) as specified at Section 14.8 (Calculating and Verifying the UTRN Check Digit); and
- shall disregard the 20th decimal digit to determine PPTD prior to undertaking any subsequent checks.

14.6.4.1.2 Using the PPTD to calculate the PTUT

PTUT shall take the value of PPTD minus 7,394,156,990,786,306,048.

The Device shall interpret the resulting unsigned integer according to Table 14.3.2.

14.6.4.1.3 Verifying PTUT subclass category

The Device shall carry out the checks specified in Section 14.3.9.

14.6.4.1.4 Verifying against the maximum credit values

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 Counter[45]

The Originator Counter shall be derived by:

1. creating four 32 bit signed integer variables p, q, r and s;
2. setting p = the numeric value of the 10 least significant bits of Highest UTRN Counter;
3. setting q = (the numeric value Highest UTRN Counter) – p;
4. setting r = the numeric value of PTUT Truncated Originator Counter;
5. if $r < (p - 2^9)$ then setting $s = (r + 2^{10})$ else if $r > (p + 2^9)$ then setting $s = (r - 2^{10})$ else setting s = r;
6. setting Originator Counter equal to $((q + s) * 2^{32})$

14.6.4.1.6 Verifying the Originator Counter

The Device shall verify the Originator Counter against the UTRN Counter Cache according to Section 14.3.6.

14.6.4.1.7 Deriving the Message Identifier

The Device shall derive the Message Identifier by:

- setting the Business Target ID to the Entity Identifier in the Key Agreement Security Credentials it holds for the Trust Anchor Cell with Remote Party Role as `supplier` and cell usage of `prePaymentTopUp`;
- setting the Business Originator ID to its own Entity Identifier; and
- setting Message Identifier to the concatenation Business Originator ID || Business Target ID || 0x02 || Originator Counter.

14.6.4.1.8 Validating the PTUT Supplier MAC

The Device shall validate the PTUT Supplier MAC according to Section 14.3.5.

14.6.5 Response Construction

The Device shall first update the UTRN Counter Cache according to Section 14.3.7.

Where the Device is an ESME, the Device shall construct, and send via its HAN interface, a Response message complying with the requirements of Use Case CS01a, using a Message Identifier as specified in Section 14.6.4.1.7, and where the Originator Counter is as derived by the calculations in Section 14.6.4.1.5.

Where the Device is a GSME, the Device shall construct, and send via its HAN interface, a Response message complying with the requirements of Use Case CS01b, using Message Identifier as specified in Section 14.6.4.1.7, and where the Originator Counter is as derived by the calculations in Section 14.6.4.1.5.

[45] This derivation places a practical limit on the maximum increment between issued sequentially UTRN Counters. An increment of greater than (2^9 - 1) between a UTRN Counter and the next one issued will cause this derivation to be inaccurate

14.7 Applying a Prepayment Top Up to an ESME or GSME with consumer entry of a numeric code on a PPMID

14.7.1 Description

This Section covers the application of a Prepayment Top Up that has been bought for a specific GSME or ESME to that GSME or ESME in the case where the consumer enters the corresponding UTRN on a PPMID on the same SMHAN.

The Use Case covering the Command is referenced in Section 14.7.4.1.2, The Use Case covering the Response is referenced in Section 14.7.4.1.4.

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
Service 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

Table 14.7.2: Use Case Cross References for Prepayment Top Up through PPMID entry

14.7.3 Pre-conditions

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 Top Up 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 the PPTD by a process equivalent to the following (Verhoeff's) Algorithm[46]:

- 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;

[46] See: (1) Verhoeff, J. (1969). Error Detecting Decimal Codes (Tract 29). The Mathematical Centre, Amsterdam. doi: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. Retrieved August 26, 2011. (3) Salomon, David (2005). Coding for Data and Computer Communications. Springer. p. 56. ISBN 0-387-21245-0. Retrieved August 26, 2011

- 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 Jth digit of the 19 decimal digits of the PPTD, 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;
  - 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	0	0	1	2	3	4	5	6	7	8	9
Index	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
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

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

Table 14.8c: Setting IntDig using IntDig as a Column Index

15 Message Codes

Message Codes shall be 2 octets in length and shall take the values specified in the 'Use Case reference' tab in the Mapping Table.

For Messages specified by this GBCS, the most significant bit of the Message Code shall be 0b0.

16 Event / Alert Codes and related requirements

Italicised terms in this Section 16 shall have their meaning in the ZCL / ZSE specifications.

16.1 Introduction – informative

This Section 16 sets out how Events and Alerts are handled. The Technical Specifications define when Events occur and whether these Events are logged (in an Event Log) and whether sent as an Alert via the HAN / WAN.

Table 16.2 defines Event Codes for events defined in the Technical Specifications. It also indicates whether, as per the Technical Specifications, 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 of the applicable Device Type must support;
- Mandated conditional – Devices of the applicable Device Type 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.

An SAPC shall support the Alerts, and support corresponding logging requirements, for rows in Table 16.2 marked 'mandatory' in the column titled 'SAPC support'. For clarity, 'ECS100 Command not supported by Device' is marked as 'mandatory' and must be supported by an SAPC unless the SAPC supports all Use Cases required of an ESME which is a Single Element Electricity Metering Equipment, with its SMETS meaning.

An SAPC, where it has a Boost Function, with its SMETS meaning, shall support the Alerts, and support corresponding logging requirements, for rows in Table 16.2 marked 'mandatory where the SAPC has a Boost Function, with its SMETS meaning' in the column titled 'SAPC support'.

An SAPC shall support, for each group, at least one of the Alerts, and support corresponding logging requirements, for rows in Table 16.2 marked 'mandatory – conditional group N' in the 'SAPC support' column, where 'N' is a number.

An SAPC shall additionally support the Alerts, and corresponding logging requirements, for rows in Table 16.2 marked 'optional' in the 'SAPC support' column, where the SAPC supports the corresponding SMETS functionality.

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;
- Sections 11.2 and 13.3 detail the Message construction for the Alerts with ASN.1 Payload; and
- Section 9.2.2 details the Message construction for future dated Alerts.

16.1.3 Event Behaviour

Detail on Event behaviour can be found in the Technical Specifications using the relevant SMETS and CHTS reference in Table 16.2.

16.2 Event and Alert Codes

Table 16.2 lists the valid Event and Alert Codes, and sets out their requirements.

SECMP0007-DP164-Markup___Table_16-2 GBCS v4.1 Table 16.2.xlsx

Field Code Changed

Table 16.2: Event and Alert Codes

16.3 Event Logs

Only GSME, ESME, SAPC, CHF and GPF have Event Logs. The requirement set out in Table 16.2 to log entries into Event Logs only applies to GSME, SAPC, ESME, CHF and GPF as follows:

- Event Log (GSME, SAPC, ESME, CHF and GPF);
- Security Event Log (GSME, SAPC, ESME, CHF and GPF);
- Power Event Log (ESME and SAPC where the SAPC supports the corresponding SMETS functionality); and
- Auxiliary Controller Event Log (SAPC and ESME).

Use Cases to read logs (all) and clear logs (event logs only) are detailed in the Mapping Table.

16.4 Requirements

Event / Alert codes shall be 2 octets in length and shall take the values specified in Table 16.2. As per the Technical Specifications, all Alerts, Event Log entries, Security Log entries, Power Event Log entries and Auxiliary Controller Event Log entries shall contain a UTC date time stamp, in addition to the Event / Alert code. For Event / Alert Codes beginning 0x80 or

0x81 in Table 16.2 which relate to a GSME or ESME, the requirements in Table 16.2 in relation to:

- sending Alerts;
- notifying events to other Devices to which it is connected by means of a Communications Link; and
- logging events in the Event Log or Power Event Log,

shall be those configured in ESME, SAPC and GSME prior to installation. Prior to installation, all such Event / Alert Codes shall be configured not to sound an Alarm. For clarity, Event / Alert Codes beginning 0x8F and any applying on Devices other than ESME / GSME / SAPC are not configurable and Table 16.2 specifies the fixed settings. Once installed, the configurations relating to sending Alerts, notifying other Devices, logging and sounding Alarms may be modified using the event and Alert configuration Use Cases.

GSME shall reject any ZSE *SetEventConfiguration* command containing an *Event ID* in the *Event Configuration Payload* with 0x8F in the most significant octet, to ensure Critical Alerts are always configured on. For clarity, the ESME Alert Configuration Use Cases, which may also be supported by SAPC, do not allow for Alert Codes starting 0x8F.

SAPC is not required to support the Event / Alert configuration Use Cases. Where SAPC does not support such Use Cases, the settings required by Table 16.2 shall remain unchanged, including for any Event / Alert Codes beginning 0x80 or 0x81.

As specified in Table 16.2 by way of 'x' in a cell, `deviceType` (and for ESME, variant of ESME) shall determine which Alerts a device shall issue and which Event Log and Security Log entries it shall record. Where `deviceType` = 0x04 (`HCALCS`) or 0x05 (`PPMID`), this Section 16 only requires the sending of Alerts, since neither Device type is required to have either an Event Log or a Security Log.

Where an Alert and a Log entry have the same trigger in a Device, the Device shall record the same UTC date time stamp and the same Event / Alert code in both.

The Remote Party to which an Alert containing a specific Event Code is addressed shall be determined by the Remote Party Role as specified in Table 16.2. Where the Remote Party Role is stated as 'Supplier (not CHF) or WAN Provider (CHF only)', the Alert shall be addressed:

- to the WAN Provider if `deviceType` = 0x02 (CHF); or
- to the Supplier for all other `deviceType` values.

Where the Remote Party Role is stated as 'Transitional Change of Supplier or Supplier (not CHF) or WAN Provider (CHF only)', the Alert shall be addressed to the Transitional Change of Supplier if the Alert is as a result of a `supplierByTransCoS` CS02b Update Security Credentials Command; otherwise is shall be addressed:

- to the WAN Provider if deviceType = 0x02 (CHF); or
- to the Supplier for all other deviceType values.

Where a Use Case is specified in Table 16.2 the corresponding Alert shall be constructed according to the specified Use Case. Where no Use Case is specified the Alert shall be constructed according to Section 7.

Where an Alert has two recipient roles identified, the Device shall place the Entity ID of the Supplier in the Business Target ID field and the Entity ID of the other recipient in the Supplementary Remote Party ID field.

For any Event Log entries relating to Event Codes 0x8161 and 0x8162, the Device shall record the commands input on the User Interface by including the User Interface Command Code in the Event Log entry as defined in Table 16.4.

Where an SAPC has a User Interface to support SMETS requirements, for any Event Log entries relating to Event Codes 0x8161 and 0x8162, the SAPC shall record the Commands input on the User Interface by including the User Interface Command Code in the Event Log entry as defined in Table 16.4, as if the SAPC were an ESME.

User Interface Command Code	User Interface Command (from SMETS)	GSME	ESME	Additionally for ESME with ALCS	Additionally for 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		
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 ALCS [1]			x	
0x0014	Test ALCS [2]			x	
0x0015	Test ALCS [3]			x	
0x0016	Test ALCS [4]			x	
0x0017	Test ALCS [5]			x	
0x0018	Test Valve	x			
0x0019	Reset Remaining Battery Capacity	x			
0x001A	Find and Join SMHAN	x	x	x	x

Table 16.4: User Interface Command Codes by Device

For any Event Log entries relating to Event Codes 0x8154 and 0x8155, the Device shall record the Commands received on the Network Interface by including the Message Code in the Event Log. For clarity, such entries are only required where the command received has a Message Code specified in Table 20.

Where a log entry is required to have data additional to the Alert Code and date-time stamp, that additional data shall be recorded in the 'otherInformation' field of that log entry.

For Alerts with Alert Code 0x81A0, the Device shall structure and populate the Message according to the requirements of Table 16.2 and:

- an ESME shall create a DLMS COSEM Alert Payload, which shall be as defined in Table 7.2.9c, where the Use Case specific additional content contains the concatenation 0x09 || 0x02 || otherInformation; or

- a GSME shall create a GBZ Alert Payload, which shall be as defined in Table 7.2.10c, where the Use Case specific additional content is otherInformation.

The Device shall populate the 'otherInformation' field with one of the following two octet long octet string values:

- 0x0000 (other);
- 0x0001 (Error Non Volatile Memory);
- 0x0002 (Error Program Execution);
- 0x0003 (Error Program Storage);
- 0x0004 (Error RAM);
- 0x0005 (Error Unexpected Hardware Reset);
- 0x0006 (Error Watchdog);
- 0x0007 (Error Metrology Firmware Verification Failure);
- 0x0008 (Error Measurement Fault); or
- 0x0009 (Unspecified Smart Meter Operational Integrity Error).

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 19.3). A Remote Party Role in that table 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.

18 Message Templates

18.1 GBZ and ZSE Message Templates

Message Templates for GBZ Use Cases are detailed in the embedded Use Cases, Section 19.3. These Message Templates are derived from the Mapping Table, and shall be complied with in the construction and population of all such Messages.

18.1.1 Message Templates for ZSE commands between ESME and HCALCS

Requirements in this Section 18.1.1, and sub-Sections of it, shall apply to an SAPC as if it were an ESME.

18.1.1.1 ZSE Load Control Event command

The ZSE Load Control Event command shall be sent by an ESME:

- following successful authentication of a Command with Message Code 0x011E (so 'ECS47a Set Auxiliary Controller [n] State'), where that command requires an HCALCS to change its state;
- to control an HCALCS according to the Auxiliary Controller Calendar; or
- as required by Section 18.1.1.3.

In executing this command, the ESME shall send the ZCL Load Control Event command to the HCALCS identified in that Command / in that Get Scheduled Events command / in that entry of the Auxiliary Controller Calendar (as determined by the requirements of Section 7.3.6.1) with:

- the values of each field populated in the ZCL Load Control Event command as specified in Table 18.1.1.1;
- the 'Duration in Minutes' field set according to the respective triggers above calculated as per SMETS;
- the 'Duty Cycle' field set to 0x00 or to 0x64, determined according to the requirements of Section 7.3.6.1.

The recipient HCALCS shall interpret the value in Duty Cycle accordingly.

Whenever it sends a *Load Control Event* command, the ESME shall add an entry to the Auxiliary Controller Event Log where:

- hanCommandID is set to the value of the *Issuer Event ID* in the command;
- outcome is set to 0x00 ('Outcome not known'); and
- switchNumberAndAction is set to the concatenation 0x1'n' (where 'n' is the value of Auxiliary Controller [n] for the HCALCS in question) || Duty Cycle in the command.

On successful authentication of such a ZCL command, the recipient HCALCS shall respond with a *Report Event Status* ZCL command populated as per Table 18.1.1.4, with Event Status set to:

- 0x02 ('Event started'), if the command was successfully executed; or
- 0xFE ('Load Control Event command Rejected'), if the command was not successfully executed.

Whenever it receives a *Report Event Status* command, the ESME shall add an entry to the Auxiliary Controller Event Log where:

- hanCommandID is set to the value of the *Issuer Event ID* in the command;
- outcome is set to:
  - (a) 0x01 ('Success') if Event Status is 0x02 ('Event started') in the command; or
  - (b) 0x02 ('Failure') if Event Status is 0xFE ('Load Control Event command Rejected'), in the command; and
- switchNumberAndAction is set to the concatenation 0x1'n' (where 'n' is the value of Auxiliary Controller [n] for the HCALCS in question) || Duty Cycle Applied in the command.

After the 'Duration In Minutes' specified in such a Load Control Event command has elapsed according to the HCALCS timer, the HCALCS shall send to the ESME a Get Scheduled Events command in accordance with Table 18.1.1.3. For clarity, an HCALCS may additionally send a Get Scheduled Events command to the ESME at any time.

Element	Meaning	Value	Octets
ZCL header			
Frame control	Cluster-specific; not manufacturer specific; server-client; allow default response;	0b00001001	1
Transaction sequence number		0x00	1
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

Element	Meaning	Value	Octets
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

Table 18.1.1.1: ZSE Load Control Event command

18.1.1.2 Intentionally blank

18.1.1.3 ZSE Get Scheduled Events command

When sending a ZSE Get Scheduled Event command pursuant to section 8.5.2.1 of SMETS, an HCALCS shall populate that ZCL Command according to Table 18.1.1.3.

On authenticated receipt of a ZSE Get Scheduled Events command, the ESME shall, if the Supply State is Enabled (where those terms have their SMETS meaning) send a ZSE Load Control Event command instructing the HCALCS whether it is to be open or closed, and for how long it is to be in that state, as required by SMETS.

Otherwise, the ESME shall send a ZSE Default Response command with Status = NOT_FOUND.

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 Events command

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

Element	Meaning	Value	Octets
Command identifier	Report Event Status	0x00	1
ZCL payload			
Issuer Event ID (UINT32)	Set to the Issuer Event ID from the corresponding ZSE Load Control Event 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)	Fixed value	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

18.2 DLMS COSEM Message Templates

Table 18.2 contains Message Templates for all Use Case with DLMS COSEM payloads. These Message Templates are derived from the Mapping Table, and shall be complied with in the construction and population of all such Messages.

SECMP0007-DP164-Markup___18-2___Us GBCS v4.1 Table 18.2 DLMS COSEM Messa

Field Code Changed

Table 18.2: DLMS COSEM Message Templates

18.2.1 Encoding

Italicised terms in this Section 18.2.1 shall have their DLMS COSEM meaning.

Requirements in this Section 18.2.1, and sub-Sections of it, shall, where the SAPC supports the corresponding SMETS functionality, apply to an SAPC as if it were an ESME.

18.2.1.1 Compact array encoding

The Blue Book definition of *attribute 2* of *Profile Generic* objects may be interpreted as requiring '*entry'* to be a *structure* containing a single choice from the DLMS data types. The GBCS interprets it as meaning that '*entry'* is a *structure* that can contain multiple choices of DLMS data types. These choices vary between instances of Profile Generic object. To identify these different structures, the naming convention 'entry_nameOfStructure' is used.

The GBCS uses the *compact-array* data type in attribute 2 of *Profile Generic* objects. Table 18.2.1.1 details the derivation of the *contents-description* element within the *compact-array structure* for the structures used in the *Profile Generic* objects required by this GBCS. These 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 {	0x02	0x02		0x1302020606
timestamp: double-long-unsigned,	0x06			
dlValue: double-long-unsigned	0x06			
}				
entry_enumValueLogEntry::= structure {	0x02	0x02		0x1302020616
timestamp: double-long-unsigned,	0x06			
enumValue: enum	0x16			
}				
entry_eventLogEntry12::= structure {	0x02	0x03		0x130203061209
timestamp: double-long-unsigned,	0x06			
logCode: long-unsigned,	0x12			
otherInformation: octet-string(12)	0x09			
}				
entry_powerLogEntry::= structure {	0x02	0x03		0x130203061206
timestamp: double-long-unsigned,	0x06			
logCode: long-unsigned,	0x12			
otherInformation: double-long-unsigned	0x06			
}				
entry_eventLogEntry8::= structure {	0x02	0x03		0x130203061209
timestamp: double-long-unsigned,	0x06			

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		0x1302070606010030060100080601000806010008060100080601000806 (single element) or 0x13020906060601003006010004060100080601000806010008060100080601000806 (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	0x0030	0x06	
secondaryTariffTOURegisterValues: array double-long-unsigned, [[MAY NOT BE PRESENT]]	0x01	0x0004	0x06	
tariffTOUBlock1RegisterValues: array double-long-unsigned,	0x01	0x0008	0x06	
tariffTOUBlock2RegisterValues: array double-long-unsigned,	0x01	0x0008	0x06	
tariffTOUBlock3RegisterValues: array double-long-unsigned,	0x01	0x0008	0x06	

Structure definition	*Tag*	*Number of entries (structures and arrays only)*	*Tag of entries in array*	*contents-description for compact-array*
tariffTOUBlock4RegisterValues: array double-long-unsigned	0x01	0x0008	0x06	
}				
entry_billingCalendarOnSetModeOrTariffLogEntry::= structure{	0x02	0x0D or 0x0F		0x13020D06060100300601000806010008060100080601000806050505050505 (single element) or 0x13020F06060601003006010004060100080601000806100080601000806050505050505 (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	0x0030	0x06	
secondaryTariffTOURegisterValues: array double-long-unsigned, [[MAY NOT BE PRESENT]]	0x01	0x0004	0x06	
tariffTOUBlock1RegisterValues: array double-long-unsigned,	0x01	0x0008	0x06	
tariffTOUBlock2RegisterValues: array double-long-unsigned,	0x01	0x0008	0x06	
tariffTOUBlock3RegisterValues: array double-long-unsigned,	0x01	0x0008	0x06	
tariffTOUBlock4RegisterValues: array double-long-unsigned	0x01	0x0008	0x06	
emergencyCreditBalanceValue: double-long,	0x05			
meterBalanceValue: double-long,	0x05			
paymentDebtRegisterValue: double-long,	0x05			

Structure definition	*Tag*	*Number of entries (structures and arrays only)*	*Tag of entries in array*	*contents-description for compact-array*
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		0x1302070606010030060100080601000806010008060100080601000806 (single element) or 0x1302090606060100300601000 4

Structure definition	*Tag*	*Number of entries (structures and arrays only)*	*Tag of entries in array*	*contents-description for compact-array*
				060100080601000806010008060100080601 000806 (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	0x0030	0x06	
secondaryTariffTOURegisterValues: array double-long-unsigned, [[MAY NOT BE PRESENT]]	0x01	0x0004	0x06	
tariffTOUBlock1RegisterValues: array double-long-unsigned,	0x01	0x0008	0x06	
tariffTOUBlock2RegisterValues: array double-long-unsigned,	0x01	0x0008	0x06	
tariffTOUBlock3RegisterValues: array double-long-unsigned,	0x01	0x0008	0x06	
tariffTOUBlock4RegisterValues: array double-long-unsigned	0x01	0x0008	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_twoDlValueLogEntry::= structure {	0x02	0x03		0x130203060606
timestamp: double-long-unsigned,	0x06			

Structure definition	Tag	Number of entries (structures and arrays only)	Tag of entries in array	contents-description for compact-array
dlValue: double-long-unsigned,	0x06			
dlValue2: double-long-unsigned	0x06			
}				
entry_auxiliaryControllerLogEntry::= structure {	0x02	0x04		0x13020406121606
timestamp: double-long-unsigned,	0x06			
switchNumberAndAction: long-unsigned,	0x12			
outcome: enum,	0x16			
hANCommandID: double-long-unsigned	0x06			
}				

Table18.2.1.1: derivation of the *contents-description* element within the *compact-array structure*

18.2.1.2 Values of the credit_charge_configuration attribute of Account (Class ID 111) objects

There are three SMETS parameters required for all Set Payment Mode Use Cases:

- Payment Mode, being Credit or Prepayment;
- Suspend Debt Emergency, being True or False and only being relevant when Payment Mode = Prepayment; and
- Suspend Debt Disabled, being True or False and only being relevant when Payment Mode = Prepayment.

Note that Disablement Threshold can also be set through the 'Set Payment Mode to Prepayment' Use Case.

The combination of these values determines, and is reflected in, the five possible values in the *credit_charge_configuration* attribute of the Account objects.

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 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 020309060000130A00FF09060000131400FF0403E0 020309060000130A00FF09060000131404FF0403E0	44
Prepayment	True	True	0x010D 020309060000130A00FF09060000131400FF0403E0 020309060000130A00FF09060000131404FF0403E0 020309060000130A00FF09060000131401FF0403C0 020309060000130A00FF09060000131402FF0403C0 020309060000130A00FF09060000131403FF0403E0 020309060000130A01FF09060000131403FF0403E0 020309060000130A01FF09060000131400FF0403E0 020309060000130A01FF09060000131404FF0403E0 020309060000130A01FF09060000131401FF0403C0 020309060000130A01FF09060000131402FF0403C0 020309060000130A02FF09060000131404FF0403E0 020309060000130A02FF09060000131401FF0403E0 020309060000130A02FF09060000131402FF0403E0	275
Prepayment	True	False	0x010D 020309060000130A00FF09060000131400FF0403E0 020309060000130A00FF09060000131404FF0403E0 020309060000130A00FF09060000131401FF0403E0 020309060000130A00FF09060000131402FF0403E0 020309060000130A00FF09060000131403FF0403E0 020309060000130A01FF09060000131403FF0403E0 020309060000130A01FF09060000131400FF0403E0 020309060000130A01FF09060000131404FF0403E0 020309060000130A01FF09060000131401FF0403E0 020309060000130A01FF09060000131402FF0403E0 020309060000130A02FF09060000131404FF0403E0 020309060000130A02FF09060000131401FF0403E0 020309060000130A02FF09060000131402FF0403E0	275

Payment Mode	Suspend Debt Emergency	Suspend Debt Disabled	Value of credit_charge_configuration attribute	Length in octets
Prepayment	False	True	0x010A 020309060000130A00FF09060000131400FF0403E0 020309060000130A00FF09060000131404FF0403E0 020309060000130A00FF09060000131401FF0403C0 020309060000130A00FF09060000131402FF0403C0 020309060000130A00FF09060000131403FF0403E0 020309060000130A01FF09060000131403FF0403E0 020309060000130A01FF09060000131400FF0403E0 020309060000130A01FF09060000131404FF0403E0 020309060000130A01FF09060000131401FF0403C0 020309060000130A01FF09060000131402FF0403C0	212
Prepayment	False	False	0x010A 020309060000130A00FF09060000131400FF0403E0 020309060000130A00FF09060000131404FF0403E0 020309060000130A00FF09060000131401FF0403E0 020309060000130A00FF09060000131402FF0403E0 020309060000130A00FF09060000131403FF0403E0 020309060000130A01FF09060000131403FF0403E0 020309060000130A01FF09060000131400FF0403E0 020309060000130A01FF09060000131404FF0403E0 020309060000130A01FF09060000131401FF0403E0 020309060000130A01FF09060000131402FF0403E0	212

Table 18.2.1.2a: allowable values for the *credit_charge_configuration* attribute for all ESME (except Twin Element variant)

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 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 020309060000130A00FF09060000131400FF0403E0 020309060000130A00FF09060000131404FF0403E0	65

Payment Mode	Suspend Debt Emergency	Suspend Debt Disabled	Value of credit_charge_configuration attribute	Length in octets
			020309060000130A00FF09060000131405FF0403E0	
Prepayment	True	True	0x010F 020309060000130A00FF09060000131400FF0403E0 020309060000130A00FF09060000131404FF0403E0 020309060000130A00FF09060000131401FF0403C0 020309060000130A00FF09060000131402FF0403C0 020309060000130A00FF09060000131403FF0403E0 020309060000130A01FF09060000131403FF0403E0 020309060000130A01FF09060000131400FF0403E0 020309060000130A01FF09060000131404FF0403E0 020309060000130A01FF09060000131401FF0403C0 020309060000130A01FF09060000131402FF0403C0 020309060000130A02FF09060000131404FF0403E0 020309060000130A02FF09060000131401FF0403E0 020309060000130A02FF09060000131402FF0403E0 020309060000130A00FF09060000131405FF0403E0 020309060000130A01FF09060000131405FF0403E0	317
Prepayment	True	False	0x010F 020309060000130A00FF09060000131400FF0403E0 020309060000130A00FF09060000131404FF0403E0 020309060000130A00FF09060000131401FF0403E0 020309060000130A00FF09060000131402FF0403E0 020309060000130A00FF09060000131403FF0403E0 020309060000130A01FF09060000131403FF0403E0 020309060000130A01FF09060000131400FF0403E0 020309060000130A01FF09060000131404FF0403E0 020309060000130A01FF09060000131401FF0403E0 020309060000130A01FF09060000131402FF0403E0 020309060000130A02FF09060000131404FF0403E0 020309060000130A02FF09060000131401FF0403E0 020309060000130A02FF09060000131402FF0403E0	317

Payment Mode	Suspend Debt Emergency	Suspend Debt Disabled	Value of credit_charge_configuration attribute	Length in octets
			020309060000130A00FF09060000131405FF0403E0 020309060000130A01FF09060000131405FF0403E0	
Prepayment	False	True	0x010C 020309060000130A00FF09060000131400FF0403E0 020309060000130A00FF09060000131404FF0403E0 020309060000130A00FF09060000131401FF0403C0 020309060000130A00FF09060000131402FF0403C0 020309060000130A00FF09060000131403FF0403E0 020309060000130A01FF09060000131403FF0403E0 020309060000130A01FF09060000131400FF0403E0 020309060000130A01FF09060000131404FF0403E0 020309060000130A01FF09060000131401FF0403C0 020309060000130A01FF09060000131402FF0403C0 020309060000130A00FF09060000131405FF0403E0 020309060000130A01FF09060000131405FF0403E0	254
Prepayment	False	False	0x010C 020309060000130A00FF09060000131400FF0403E0 020309060000130A00FF09060000131404FF0403E0 020309060000130A00FF09060000131401FF0403E0 020309060000130A00FF09060000131402FF0403E0 020309060000130A00FF09060000131403FF0403E0 020309060000130A01FF09060000131403FF0403E0 020309060000130A01FF09060000131400FF0403E0 020309060000130A01FF09060000131404FF0403E0 020309060000130A01FF09060000131401FF0403E0 020309060000130A01FF09060000131402FF0403E0 020309060000130A00FF09060000131405FF0403E0 020309060000130A01FF09060000131405FF0403E0	254

Table 18.2.1.2b: allowable values for the *credit_charge_configuration* attribute for all ESME (Twin Element variant)

18.2.1.3 Deriving the values of the credit_charge_configuration attribute of Account (Class ID 111) objects – informative

This Section explains the derivation of the five values of this attribute that an ESME can accept.

The *credit_charge_configuration* attribute encoding is shown in Table 18.2.1.3a.

Component	Hex value	Length in octets	Notes
credit_charge_configuration			
Tag	0x01	1	tag for array
Length	Variable	1	entries in array
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	

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 concatenation:

0x02030906 || credit object OBIS code || 0x0906 || charge object OBIS code || 0x0403 || collection bit string || 0b00000

The meaning of each *credit_charge_configuration_element* is that this charge can be collected from this credit object, except in possible meter states specified by the *collection_configuration* bit string.

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
AccumulatedDebt	0-0:19.10.2.255	0x0000130A02FF	Prepayment
EmergencyCreditBalance	0-0:19.10.1.255	0x0000130A01FF	Prepayment

Table 18.2.1.3b: Class 112 Credit objects

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 relevant in Credit Mode.

SMET Reference Component	*OBIS Code (decimal)*	*OBIS Code (hexadecimal)*	*Payment Mode*
DebtRecoveryRates[1]	0-0:19.20.1.255	0x0000131401FF	Prepayment
DebtRecoveryRates[2]	0-0:19.20.2.255	0x0000131402FF	Prepayment
DebtRecoveryPerPayment	0-0:19.20.3.255	0x0000131403FF	Prepayment
SecondaryTariffTOUPriceMatrix (Twin element ESME only)	0-0:19.20.5.255	0x0000131405FF	Prepayment and Credit
StandingCharge	0-0:19.20.4.255	0x0000131404FF	Prepayment and Credit
TariffBlockPriceMatrixTOU	0-0:19.20.0.255	0x0000131400FF	Prepayment and Credit

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 collected from a credit dependent on ESME state.

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 limiting periods).

In Credit Mode, collection continues in all states, so the value of all three bits is always 0b1.

In Prepayment Mode, *collection_configuration* is set according to Suspend Debt Disabled (affects Bit 0) values, and the pairing of charge and credit object.

Suspend Debt Emergency being True means that DebtRecoveryRates[1..2] and StandingCharge are collected from AccumulatedDebt rather than EmergencyCreditBalance, when Emergency Credit is in use, so Suspend Debt Emergency is specified by way of pairing charge and credit objects accordingly. Note that Bit 2 of *collection_configuration* shall always be fixed at 0b1.

Suspend Debt Disabled being True means that DebtRecoveryRates[1..2] are no longer collected when the supply is disabled due to lack of credit.

Table 18.2.1.3d sets out the *credit_charge_configuration_element* array entries in Credit Mode.

Tag & Length	Credit object	Tag & Length	Charge object	Tag & Length	Collection bit string	trailing_ bits	Array entry
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000131400FF (TariffBlockPriceMatrixTOU)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000131400FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000131404FF (StandingCharge)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000131404FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000131405FF (SecondaryTariffTOUPriceMatrix (Twin element ESME only))	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000131405FF0403E0

Table 18.2.1.3d: Class 113 Charge objects

When the mode is set as in Table 18.2.1.3e:

Payment Mode	Suspend Debt Emergency	Suspend Debt Disabled
Prepayment	False	False

Table 18.2.1.3e: Prepayment states

the *credit_charge_configuration_element* array entries are as per Table 18.2.1.3f.

Tag & Length	Credit object	Tag & Length	Charge object	Tag & Length	Collection bit string	trailing_ bits	Array entry
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000131400FF (TariffBlockPriceMatrixTOU)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000131400FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000131404FF (StandingCharge)	0x0403	0b111	0b00000	0x020309060000130A00FF09060000131404FF0403E0

Tag & Length	Credit object	Tag & Length	Charge object	Tag & Length	Collection bit string	trailing_ bits	Array entry
					(collectable in all circumstances)		
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000131401FF (DebtRecoveryRates[1])	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000131401FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000131402FF (DebtRecoveryRates[2])	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000131402FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000131403FF (DebtRecoveryPerPayment)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000131403FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000131403FF (DebtRecoveryPerPayment)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000131403FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000131400FF (TariffBlockPriceMatrixTOU)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000131400FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000131404FF (StandingCharge)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000131404FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000131401FF (DebtRecoveryRates[1])	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000131401FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000131402FF (DebtRecoveryRates[2])	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000131402FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000131405FF (SecondaryTariffTOUPriceMatrix (Twin element ESME only))	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000131405FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000131405FF (SecondaryTariffTOUPriceMatrix (Twin element ESME only))	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000131405FF0403E0

Table 18.2.1.3f: *credit_charge_configuration_element* array entries

Note that, as per SMETS, the value of MeterBalance determines whether charges are collected from EmergencyCreditBalance or MeterBalance.

When the mode is set as in Table 18.2.1.3g:

Payment Mode	Suspend Debt Emergency	Suspend Debt Disabled
Prepayment	False	True

Table 18.2.1.3g: Prepayment states

the *credit_charge_configuration_element* array entries are as per Table 18.2.1.3h.

Tag & Length	*Credit object*	*Tag & Length*	*Charge object*	*Tag & Length*	*Collection bit string*	*trailing_ bits*	*Array entry*
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000131400FF (TariffBlockPriceMatrixTOU)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000131400FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000131404FF (StandingCharge)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000131404FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000131401FF (DebtRecoveryRates[1])	0x0403	0b110 (do not collect when supply is disabled due to no credit)	0b00000	0x020309060000130A00FF09060000131401FF0403C0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000131402FF (DebtRecoveryRates[2])	0x0403	0b110 (do not collect when supply is disabled due to no credit)	0b00000	0x020309060000130A00FF09060000131402FF0403C0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000131403FF (DebtRecoveryPerPayment)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000131403FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000131403FF (DebtRecoveryPerPayment)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000131403FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000131400FF (TariffBlockPriceMatrixTOU)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000131400FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000131404FF (StandingCharge)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000131404FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000131401FF (DebtRecoveryRates[1])	0x0403	0b110 (do not collect when supply is disabled due to no credit)	0b00000	0x020309060000130A01FF09060000131401FF0403C0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000131402FF (DebtRecoveryRates[2])	0x0403	0b110 (do not collect when supply is disabled due to no credit)	0b00000	0x020309060000130A01FF09060000131402FF0403C0

Tag & Length	Credit object	Tag & Length	Charge object	Tag & Length	Collection bit string	trailing_ bits	Array entry
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000131405FF (SecondaryTariffTOUPriceMatrix (Twin element ESME only))	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000131405FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000131405FF (SecondaryTariffTOUPriceMatrix (Twin element ESME only))	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000131405FF0403E0

Table 18.2.1.3h: *credit_charge_configuration_element* array entries

When the mode is set as in Table 18.2.1.3i:

Payment Mode	Suspend Debt Emergency	Suspend Debt Disabled
Prepayment	True	False

Table 18.2.1.3i: Prepayment states

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	0x0000131400FF (TariffBlockPriceMatrixTOU)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000131400FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000131404FF (StandingCharge)	0x0403	0b111 (collectable in all circumstances)	0b0000	0x020309060000130A00FF09060000131404FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000131401FF (DebtRecoveryRates[1])	0x0403	0b111 (collectable in all circumstances)	0b0000	0x020309060000130A00FF09060000131401FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000131402FF (DebtRecoveryRates[2])	0x0403	0b111 (collectable in all circumstances)	0b0000	0x020309060000130A00FF09060000131402FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000131403FF (DebtRecoveryPerPayment)	0x0403	0b1110 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000131403FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000131403FF (DebtRecoveryPerPayment)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000131403FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000131400FF (TariffBlockPriceMatrixTOU)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000131400FF0403E0

Tag & Length	Credit object	Tag & Length	Charge object	Tag & Length	Collection bit string	trailing_ bits	Array entry
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000131404FF (StandingCharge)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000131404FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000131401FF (DebtRecoveryRates[1])	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000131401FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000131402FF (DebtRecoveryRates[2])	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000131402FF0403E0
0x02030906	0x0000130A02FF (AccumulatedDebt)	0x0906	0x0000131404FF (StandingCharge)	0x0403	0b111 (collect in Emergency Credit period – see note at bottom of table)	0b00000	0x020309060000130A02FF09060000131404FF0403E0
0x02030906	0x0000130A02FF (AccumulatedDebt)	0x0906	0x0000131401FF (DebtRecoveryRates[1])	0x0403	0b111 (collect in Emergency Credit period – see note at bottom of table)	0b00000	0x020309060000130A02FF09060000131401FF0403E0
0x02030906	0x0000130A02FF (AccumulatedDebt)	0x0906	0x0000131402FF (DebtRecoveryRates[2])	0x0403	0b111 (collect in Emergency Credit period – see note at bottom of table)	0b00000	0x020309060000130A02FF09060000131402FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000131405FF (SecondaryTariffTOUPriceMatrix (Twin element ESME only))	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000131405FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000131405FF (SecondaryTariffTOUPriceMatrix (Twin element ESME only))	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000131405FF0403E0

Table 18.2.1.3j: *credit_charge_configuration_element* array entries

Note that, as per SMETS, charges shall only accrue to AccumulatedDebt in Emergency Credit periods.

When the mode is set as in Table 18.2.1.3k:

Payment Mode	Suspend Debt Emergency	Suspend Debt Disabled
Prepayment	True	True

Table 18.2.1.3k Prepayment states

the *credit_charge_configuration_element* array entries are as per Table 18.2.1.3l.

Tag & Length	*Credit object*	*Tag & Length*	*Charge object*	*Tag & Length*	*Collection bit string*	*trailing_ bits*	*Array entry*
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000131400FF (TariffBlockPriceMatrixTOU)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000131400FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000131404FF (StandingCharge)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000131404FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000131401FF (DebtRecoveryRates[1])	0x0403	0b110 (do not collect when supply is disabled due to no credit)	0b00000	0x020309060000130A00FF09060000131401FF0403C0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000131402FF (DebtRecoveryRates[2])	0x0403	0b110 (do not collect when supply is disabled due to no credit)	0b00000	0x020309060000130A00FF09060000131402FF0403C0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000131403FF (DebtRecoveryPerPayment)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000131403FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000131403FF (DebtRecoveryPerPayment)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000131403FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000131400FF (TariffBlockPriceMatrixTOU)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000131400FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000131404FF (StandingCharge)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000131404FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000131401FF (DebtRecoveryRates[1])	0x0403	0b110 (do not collect when supply is disabled due to no credit)	0b00000	0x020309060000130A01FF09060000131401FF0403C0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000131402FF (DebtRecoveryRates[2])	0x0403	0b110 (do not collect when supply is disabled due to no credit)	0b00000	0x020309060000130A01FF09060000131402FF0403C0
0x02030906	0x0000130A02FF (AccumulatedDebt)	0x0906	0x0000131404FF (StandingCharge)	0x0403	0b111	0b00000	0x020309060000130A02FF09060000131404FF0403E0

Tag & Length	Credit object	Tag & Length	Charge object	Tag & Length	Collection bit string	trailing_ bits	Array entry
					(collect in Emergency Credit period – see note at bottom of table)		
0x02030906	0x0000130A02FF (AccumulatedDebt)	0x0906	0x0000131401FF (DebtRecoveryRates[1])	0x0403	0b111 (collect in Emergency Credit period – see note at bottom of table)	0b00000	0x020309060000130A02FF09060000131401FF0403E0
0x02030906	0x0000130A02FF (AccumulatedDebt)	0x0906	0x0000131402FF (DebtRecoveryRates[2])	0x0403	0b111 (collect in Emergency Credit period – see note at bottom of table)	0b00000	0x020309060000130A02FF09060000131402FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000131405FF (SecondaryTariffTOUPriceMatrix (Twin element ESME only))	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000131405FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000131405FF (SecondaryTariffTOUPriceMatrix (Twin element ESME only))	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000131405FF0403E0

Table 18.2.1.3l: *credit_charge_configuration_element* array entries

Note that, as per SMETS, charges shall only accrue to AccumulatedDebt in Emergency Credit periods.

18.2.1.4 Encoding of Billing Calendar start date-time and periodicity where periodicity is monthly or shorter

Table 18.2.1.4 sets out how the components of the Billing Calendar start date-time and periodicity should be encoded where periodicity is monthly or shorter.

Component	Hex value	Length in octets	Notes
execution_ time			
Tag	0x01	1	Tag for array
Length	0x01	1	1 entry in array
execution_time_date			
Tag	0x02	1	Tag for structure

Component	Hex value	Length in octets	Notes
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
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

Table 18.2.1.4: Encoding of Billing Calendar start date-time and periodicity

18.2.1.5 Encoding of Billing Calendar start date-time and periodicity where periodicity is quarterly or longer

Where the periodicity is quarterly or longer, this Section sets out how the components of the Billing Calendar start date-time and periodicity shall be encoded in attribute 4 (execution_time) of the DLMS COSEM class 22 (OBIS 0-0:15.0.0.255 in Use Case ECS30a).

If the required periodicity is quarterly (so once every three months), the execution_time shall be encoded as per Table 18.2.1.5a.

If the required periodicity is six monthly, the execution_time shall be encoded as per Table 18.2.1.5b.

If the required periodicity is yearly, the execution_time shall be encoded as per Table 18.2.1.5c.

Note that month values are to be calculated using modulo twelve addition. For example, if quarterly billing is required and the start date-time is in October (so month 10), the value of months in the following three quarters is calculated by adding three modulo 12, so giving 1 (January), 4 (April) and 7 (July).

Component	*Hex value*	*Length in octets*	*Notes*
execution_time			
Tag	0x01	1	Tag for array
Length	0x04	1	4 entries in the array, one for each of the four quarters in each year
execution_time_date			First entry in the array
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
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,	See note	4	Month part of the start date-time, as per section 4.1.6.1 of the Blue Book
day of month,	See note	4	Day of month part of the start date-time, as per section 4.1.6.1 of the Blue Book
day of week	0xFF	1	0xFF means not specified
execution_time_date			Second entry in the array
Tag	0x02	1	Tag for structure
Length	0x02	1	2 elements in structure

Component	Hex value	Length in octets	Notes
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
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,	See note	4	Month part of the start date-time plus three modulo twelve, as per section 4.1.6.1 of the Blue Book
day of month,	See note	4	Day of month part of the start date-time, as per section 4.1.6.1 of the Blue Book
day of week	0xFF	1	0xFF means not specified
execution_time_date			Third entry in the array
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
Date			
Tag	0x09	1	5 octets in DLMS encoded date
Length	0x05	1	2 elements in structure
Value			

Component	Hex value	Length in octets	Notes
year highbyte,	0xFF	1	0xFF means not specified
year lowbyte,	0xFF	1	0xFF means not specified
month,	See note	4	Month part of the start date-time plus six modulo twelve, as per section 4.1.6.1 of the Blue Book
day of month,	See note	4	Day of month part of the start date-time, as per section 4.1.6.1 of the Blue Book
day of week	0xFF	1	0xFF means not specified
execution_time_date			Fourth entry in the array
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
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,	See note	4	Month part of the start date-time plus nine modulo twelve, as per section 4.1.6.1 of the Blue Book
day of month,	See note	4	Day of month part of the start date-time, as per section 4.1.6.1 of the Blue Book
day of week	0xFF	1	0xFF means not specified

Table 18.2.1.5a: Encoding of Billing Calendar start date-time and periodicity where periodicity is quarterly

Component	Hex value	Length in octets	Notes
execution_ time			
Tag	0x01	1	Tag for array
Length	0x02	1	2 entries in the array, one for each of the six months in each year
execution_time_date			First entry in the array
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
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,	See note	4	Month part of the start date-time, as per section 4.1.6.1 of the Blue Book
day of month,	See note	4	Day of month part of the start date-time, as per section 4.1.6.1 of the Blue Book
day of week	0xFF	1	0xFF means not specified
execution_time_date			Second entry in the array
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

Component	Hex value	Length in octets	Notes
Value	See note	4	Time part of the start date-time, as per section 4.1.6.1 of the Blue Book
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,	See note	4	Month part of the start date-time plus six modulo twelve, as per section 4.1.6.1 of the Blue Book
day of month,	See note	4	Day of month part of the start date-time, as per section 4.1.6.1 of the Blue Book
day of week	0xFF	1	0xFF means not specified

Table 18.2.1.5b: Encoding of Billing Calendar start date-time and periodicity where periodicity is six monthly

Component	Hex value	Length in octets	Notes
execution_time			
Tag	0x01	1	Tag for array
Length	0x01	1	1 entry in the array
execution_time_date			First entry in the array
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
Date			
Tag	0x09	1	5 octets in DLMS encoded date

Component	Hex value	Length in octets	Notes
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,	See note	4	Month part of the start date-time, as per section 4.1.6.1 of the Blue Book
day of month,	See note	4	Day of month part of the start date-time, as per section 4.1.6.1 of the Blue Book
day of week	0xFF	1	0xFF means not specified

Table 18.2.1.5c: Encoding of Billing Calendar start date-time and periodicity where periodicity is yearly

To detail the fixed value required in the ESME for attribute 3 of the DLMS COSEM class 22 object ((OBIS 0-0:15.0.0.255 in Use Cases ECS30a), so as to allow for a variable size of the execution_time array.

18.2.1.6 Encoding to turn daily Block Counter resetting on and off

This Section sets out how attribute 4 (execution_time) of the DLMS COSEM class 22 object (OBIS 0-0:15.1.0.255 in Use Case ECS48) is to be set, so as to either turn on or off daily resetting of Block Counters.

If daily resetting of Block Counters is required, the execution_time shall be encoded as per Table 18.2.1.6a.

If daily resetting of Block Counters is not required, the execution_time shall be encoded as per Table 18.2.1.6b. Note this uses the same 31/12/3000 date as DCC User Interface Specification to mean never execute.

Component	Hex value	Length in octets	Notes
execution_ time			
Tag	0x01	1	Tag for array
Length	0x01	1	1 entry in the array
execution_time_date			First entry in the array
Tag	0x02	1	Tag for structure
Length	0x02	1	2 elements in structure
Time			
Tag	0x09	1	Tag for structure

Component	Hex value	Length in octets	Notes
Length	0x04	1	4 octets in DLMS encoded time
Value	0x00000000	4	Midnight UTC
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	1	0xFF means not specified
day of week	0xFF	1	0xFF means not specified

Table 18.2.1.6a: Encoding of execution_time where daily resetting of Block Counters is required.

Component	Hex value	Length in octets	Notes
execution_ time			
Tag	0x01	1	Tag for array
Length	0x01	1	1 entry in the array
execution_time_date			First entry in the array
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	0x00000000	4	Midnight UTC
Date			
Tag	0x09	1	5 octets in DLMS encoded date
Length	0x05	1	2 elements in structure

Component	Hex value	Length in octets	Notes
Value			
year highbyte,	0x0B	1	Year 3000 highbyte
year lowbyte,	0xB8	1	Year 3000 lowbyte
month,	0x0C	1	Meaning December
day of month,	0x1F	1	31st
day of week	0xFF	1	0xFF means not specified

Table 18.2.1.6b: Encoding of execution_time where daily resetting of Block Counters is not required.

18.3 Illustrative command and response instantiation and DER encoding

18.3.1 Illustrative `@UpdateSecurityCredentials.CommandPayload` instantiation and its DER encoding – informative

`supplierUpdatingAllSupplierCertificates` in Table 18.3.1a is an ASN.1 structured value assignment. This specific example is where a Device's Supplier is instructing the Device to replace both the Supplier Digital Signing and Key Agreement credentials on the Device, and resetting Execution Counters. In business terms, an example of this would be at Change of Supplier.

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 values.

ASN.1	Notes

`supplierUpdatingAllSupplierCertificates CommandPayload ::=`
`{authorisingRemotePartyControl`
`{credentialsReplacementMode supplierBySupplier,`
	This message is for the supplier replacing supplier credentials

`authorisingRemotePartyTACellIdentifier`
`{trustAnchorCellRemotePartyRole supplier,`
`trustAnchorCellKeyUsage { digitalSignature}},`
	The public key to be used to check the signature on this message is the supplier digital signing key currently held by the Device.

ASN.1	Notes
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,	The new supplier's digital signing certificate …
targetTrustAnchorCell	
{trustAnchorCellRemotePartyRole supplier,	… which is to be placed in the Device's supplier, digital signature Trust Anchor Cell
trustAnchorCellKeyUsage { digitalSignature}}}	
{replacementCertificate '0B34269F123456789ABCDEF01234'H,	The new supplier's key agreement certificate…
targetTrustAnchorCell	
{trustAnchorCellRemotePartyRole supplier,	which is to be placed in the Device's supplier, key agreement Trust Anchor Cell
trustAnchorCellKeyUsage {keyAgreement}}}}	
certificationPathCertificates {'FFAABB9F123456789ABCDEF01234'H }}	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.

Table 18.3.1a: Illustrative @UpdateSecurityCredentials.CommandPayload instantiation – ASN.1 structure

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 explained in Table 18.3.1b. For these purposes, the Certificate is simply shown as an OCTET STRING.

Component	Value	Notes
CommandPayload 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:		
tag = [3] primitive;	0x83	Tag for INTEGER
length =	0x04	4 octet length INTEGER

Component	Value	Notes
contents =:	0x075bcd15	The old supplier's Execution Counter in hex
newRemotePartyFloorSeqNumber SeqNumber INTEGER:		
tag = [4] primitive;	0x84	Tag for INTEGER
length =	0x04	4 octet length INTEGER
contents =:	0x3ade68b1	The new supplier's Execution Counter in hex
replacements SEQUENCE OF:		
tag = [UNIVERSAL 16] constructed;	0x30	Tag for SEQUENCE
length =	0x36	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
trustAnchorCellKeyUsage KeyUsage BIT STRING:		
tag = [UNIVERSAL 3] primitive;	0x03	Tag for BIT STRING
length =	0x02	2 octet length BIT STRING

Component	Value	Notes
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
length =	0x0e	Length of certificate
contents =:	0xffaabb9f123456789abcdef01234	CA certificate for new supplier

Table 18.3.1b: Illustrative @UpdateSecurityCredentials.Command instantiation – DER encoding

18.3.2 Illustrative `@UpdateSecurityCredentials.ResponsePayload` instantiation and its DER encoding – informative

`supplierUpdatingAllSupplierCertificatesResponse` in Table 18.3.2a is an ASN.1 structured value assignment. This specific example is where a Device is responding successfully to a Command.

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 each of the values.

ASN.1	Notes
`supplierUpdatingAllSupplierCertificatesResponse ResponsePayload ::=`	
`{ commandAccepted NULL,`	The corresponding Command was for the Supplier replacing supplier credentials
`executionOutcome`	
`{authorisingRemotePartySeqNumber 123456789,`	This is the new supplier's counter, which the Device will now use for Protection Against Replay in relation to the supplier role
`credentialsReplacementMode supplierBySupplier,`	
`remotePartySeqNumberChanges`	
`{{otherRemotePartyRole supplier,`	
`otherRemotePartyFloorSeqNumber 987654321}`	
`},`	
`replacementOutcomes`	This outcome is for the supplier digital signing store
`{{`	
`{affectedTrustAnchorCell`	
`{ trustAnchorCellRemotePartyRole supplier,`	The old supplier's Entity Identifier
`trustAnchorCellKeyUsage { digitalSignature}},`	The KeyIdentifier for the old supplier's digital signing key
`statusCode success,`	The new supplier's Entity Identifier
`existingSubjectUniqueID '123456789ABCDEF0'H,`	The KeyIdentifier for the old supplier's digital signing key
`existingSubjectKeyIdentifier '1234567890123456'H,`	This outcome is for the supplier key agreement store
`replacingSubjectUniqueID 'FEDCBA9876543210'H,`	

ASN.1	Notes
replacingSubjectKeyIdentifier 'ABCDEABCDEABCDEA'H},	
{affectedTrustAnchorCell	
{trustAnchorCellRemotePartyRole supplier,	
trustAnchorCellKeyUsage { keyAgreement}},	
statusCode success,	
existingSubjectUniqueID '123456789ABCDEF0'H,	
existingSubjectKeyIdentifier '0987654321098765'H,	
replacingSubjectUniqueID 'FEDCBA9876543210'H,	
replacingSubjectKeyIdentifier 'FEDCBFEDCBFEDCBF'H}}}}	

Table 18.3.2a: Illustrative @UpdateSecurityCredentials.Response instantiation – ASN.1 structure

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 explained in Table 18.3.2b.

Component	Value	Notes
ResponsePayload SEQUENCE:		
tag = [UNIVERSAL 16] constructed;	0X30	Tag for SEQUENCE
length =	0X8189	Length 137
content =		
commandAccepted NULL		
tag = [UNIVERSAL 5] primitive		0X05
length =	0X00	
executionOutcome ExecutionOutcome SEQUENCE		
tag = [UNIVERSAL 16] constructed	0X30	Tag for SEQUENCE
length =	0X8184	Length 132
content =		
authorisingRemotePartySeqNumber SeqNumber INTEGER:		
tag = [UNIVERSAL 2] primitive	0x02	Tag for INTEGER
length =	0x04	4 octet length INTEGER

Component	Value	Notes
contents =	0X075BCD15	The old supplier's Execution Counter in hex
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 =		
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 Execution 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:		

Component	Value	Notes
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:		
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

Component	Value	Notes
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	
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;	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 =	0X0987654321098765	KeyIdentifier
replacingSubjectUniqueID OCTET STRING:		
tag = [UNIVERSAL 4] primitive;	0X04	Tag for OCTET STRING
length =	0X08	8 octet length of Entity Identifier

Component	Value	Notes
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.ResponsePayload instantiation – DER encoding

18.4 Cryptographic Test Vectors

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.

18.4.1 Cryptographic Calculations

Create details for three Smart Metering Entities with associated Keys and shared secrets:

An Entity called SupplierA:

--With an Entity ID:	0x12:34:56:78:9A:BC:DE:F0
--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:F0:38:7A: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:B0: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:E0:87:40:70:3B: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:78:8C: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

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:A0:37:D6:D1:78:01:97:96:E7:6E:52:55:BD:C3:A0:8E:34:6F:9F:6E:6E:7E:8F:6A:4D:55:96:2D:2F:2D:0E: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:03:0A:00:06:36:47:B2:0E:13:FE

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:C8:35:0A:DB:40:51:AC:91:3C:A4:70:CF:42:2D:8A:53:DE:8C:88:1D:BF:FE:B4:0B:A4:70:51
--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:89:0E: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:0B: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:A0: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:47: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

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:F0
--Business Target ID:	0xFF:FF:FF:FF:FF:FF:FF:FE
--Date Time:	0x
--Other Info:	0x00:B3
--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:02:03:05:00:00:00:00:05:00:00:00:00:05:00:00:00:00

--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:	0x09:01:00:00:00:00:00:00:00:01:08:12:34:56:78:9A: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:02:03:05:00:00:00:00:05:00:00:00:00:05:00:00:00:00
--The per message secret number:	603369623275390501917527158020836207998575625697561737785547464758426196177 75
--The resulting Signature in Plain Format:	0x59:04:5A:B0:F5:54:62:2B:03:60:34:0B:9D:87:B5:A2:3E:D5:72:3B:41:DE:3F:20:6E:58:CD:D1:0F:91:5B:9F:E2:E1:2E:2D:A3:63:24:78:A8:DF:67:8E:41:88:95:86:9A:C1:E5:53:18:CC:E0:4D:12:0D:2D:6B:44:DC:16:7B
--The Grouping Header:	0xDF:09:01:00:00:00:00:00:00:00:01:08:12:34:56:78:9A:BC:DE:F0:08:FF:FF:FF:FF:FF:FF:FE:00:02:00:B3:35
--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: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:02:03:05:00:00:00:00:05:00:00:00:00:05:00:00:00:00:40:59:04:5A:B0:F5:54:62:2B:03:60:34:0B:9D:87:B5:A2:3E:D5:72:3B:41:DE:3F:20:6E:58:CD:D1:0F:91:5B:9F:E2:E1:2E:2D:A3:63:24:78:A8:DF:67:8E:41:88:95:86:9A:C1:E5:53:18:CC:E0:4D:12:0D:2D:6B:44:DC:16:7B
--The KDF OtherInfo:	0x60:85:74:06:08:03:00:12:34:56:78:9A:BC:DE:F0:09:01:00:00:00:00:00:00:00:01:FF: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:	0x11:00:00:00:00:00:DF:09:01:00:00:00:00:00:00:00:01:08:12:34:56:78:9A:BC:DE:F0:08:FF: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:02:03:05:00:00:00:00:05:00:00:00:00:05:00:00:00:00:40:

	59:04:5A:B0:F5:54:62:2B:03:60:34:0B:9D:87:B5:A2:3E:D5:72:3B:41:DE:3F:20:6E:58:CD:D1:0F:91:5B:9F:E2:E1:2E:2D:A3:63:24:78:A8:DF:67:8E:41:88:95:86:9A:C1:E5:53:18:CC:E0:4D:12:0D:2D:6B:44:DC:16:7B
--The resulting MAC:	0x5D:83:2D:15:B5:7A:56:D6:20:F1:98:B3
--The MAC Header excluding the Security Header	0xDD:00:00:00:00:00:00:81:A9
--The Security Header fields:	0x11:00:00:00:00
--The resulting Message:	0xDD:00:00:00:00:00:00:81:A9:11:00:00:00:00:DF:09:01:00:00:00:00:00:00:00:01:08:12:34:56:78:9A:BC:DE:F0:08:FF: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:02:03:05:00:00:00:00:05:00:00:00:00:05:00:00:00:00:40:59:04:5A:B0:F5:54:62:2B:03:60:34:0B:9D:87:B5:A2:3E:D5:72:3B:41:DE:3F:20:6E:58:CD:D1:0F:91:5B:9F:E2:E1:2E:2D:A3:63:24:78:A8:DF:67:8E:41:88:95:86:9A:C1:E5:53:18:CC:E0:4D:12:0D:2D:6B:44:DC:16:7B:5D:83:2D:15:B5:7A:56:D6:20:F1:98:B3

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:F0
--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:	0x09:02:00:00:00:00:00:00:00:01:08:FF:FF:FF:FF:FF:FF:FF:FE:08:12:34:56:78:9A:BC:DE:F0:00:02:00:B3:12:DA:20:00:00:01:00:00:03:00:00:00:03:03:00:03:00:03:00
--The per message secret number:	70516304628910536242268029808203423496928538253156798465856112218540406998811
--The resulting Signature in Plain Format:	0xA0:1A:B6:9E:D8:A6:56:6A:B1:16:43:C7:35:82:60:A8:8A:8B:60:97:85:93:E7:6A:4E:93:19:35:85:D9:8D:9B:AD:84:38:78:F2:2E:79:4B:53:F6:F2:80:F9:F1:C8:48:D9:D3:8F:C0:50:CD:DD:58:82:75:63:E2:B0:FA:19:E2
--The Grouping Header:	0xDF:09:02:00:00:00:00:00:00:00:01:08:FF: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:FF:FE:08:12:34:56:78:9A:BC:DE:F0:00:02:00:B3:12:DA:20:00:00:01:00:00:03:00:00:00:03:03:00:03:00:03:00:40:A0:1A:B6:9E:D8:A6:56:6A:B1:16:43:C7:35:82:60:A8:8A:8B:60:97:85:93:E7:6A:4E:93:19:35:85:D9:8D:9B:AD:84:38:78:F2:2E:79:4B:53:F6:F2:80:F9:F1:C8:48:D9:D3:8F:C0:50:CD:DD:58:82:75:63:E2:B0:FA:19:E2
--The resulting Message:	0xDF:09:02:00:00:00:00:00:00:00:01:08:FF:FF:FF:FF:FF:FF:FF:FE:08:12:34:56:78:9A:BC:DE:F0:00:02:00:B3:12:DA:20:00:00:01:00:00:03:00:00:00:03:03:00:03:00:03:00:40:A0:1A:B6:9E:D8:A6:56:6A:B1:16:43:C7:35:82:60:A8:8A:8B:60:97:85:93:E7:6A:4E:93:19:35:85:D9:8D:9B:AD:84:38:78:F2:2E:79:4B:53:F6:F2:80:F9:F1:C8:48:D9:D3:8F:C0:50:CD:DD:58:82:75:63:E2:B0:FA:19:E2

Supplier A has now increased its Originator Counter by 1.

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:F0
--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:00:00:00:00:80:00:FF
--The Grouping Header:	0xDF:09:01:00:00:00:00:00:00:00:02:08:12:34:56:78:9A:BC:DE:F0:08:FF:FF:FF:FF:FF:FF:FF:FE:00:02:00:22:20
--The KDF OtherInfo:	0x60:85:74:06:08:03:00:12:34:56:78:9A:BC:DE:F0:09:01:00:00:00:00:00:00:00:02:FF: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:	0x11:00:00:00:00:00:DF:09:01:00:00:00:00:00:00:00:02:08:12:34:56:78:9A: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:03:02:02:01:09:0C:07:DF:01:05:FF:00:00:00:00:80:00:FF:00
--The resulting MAC:	0x0F:1D:D0:0D:67:45:EB:D8:E0:A6:63:A4

--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:09:01:00:00:00:00:00:00:00:02:08:12:34:56:78:9A: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:03:02:02:01:09:0C:07:DF:01:05:FF:00:00:00:00:80:00:FF:00:0F:1D:D0:0D:67:45:EB:D8:E0:A6:63:A4

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:FF:FE:08:12:34:56:78:9A:BC:DE:F0:00:02:00:22:0C
--The KDF OtherInfo:	0x60:85:74:06:08:03:00:FF:FF:FF:FF:FF:FF:FF:FE:09:02:00:00:00:00:00:00:00:02:12:34:56:78:9A: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:	0x11:00:00:00:00:00:DF:09:02:00:00:00:00:00:00:00:02:08:FF: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 resulting MAC:	0x0B:3C:1B:31:2C:EA:E9:C1:30:06:0E:29
--The MAC Header excluding the Security Header	0xDD:00:00:00:00:00:00:40
--The Security Header fields:	0x11:00:00:00:00
--The resulting Message:	0xDD:00:00:00:00:00:00:40:11:00:00:00:00:DF:09:02:00:00:00:00:00:00:00:02:08:FF:FF:FF:FF:FF:FF:FF:FE:08:12:34:56:78:9A:BC:DE:F0:00:02:00:22:0C:DA:20:00:00:02:00:00:01:00:01:02:00:00:0B:3C:1B:31:2C:EA:E9:C1:30:06:0E:29

18.4.2 Example Messages Produced

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	0x000000000000	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	0x123456789ABCDEF0	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
________invoke-id	0x000001	3
____date-time	0x00	1
____access-request-body		
____access-request-specification		

____SEQUENCE OF	0x03	1
________Request number 1		
____________access-request-action	0x03	1
____________cosem-method-descriptor		
________________class-id	0x0070	2
________________instance-id	0x0000130A00FF	6
________________method-id	0x02	1
________Request number 2		
____________access-request-action	0x03	1
____________cosem-method-descriptor		
________________class-id	0x0070	2
________________instance-id	0x0000130A01FF	6
________________method-id	0x02	1
________Request number 3		
____________access-request-action	0x03	1
____________cosem-method-descriptor		
________________class-id	0x0070	2
________________instance-id	0x0000130A02FF	6
________________method-id	0x02	1
____access-request-list-of-data		
____SEQUENCE OF	0x03	1
________Parameter for request number 1		
____________Names		
____________________Tag	0x05	1
____________________Value	0x00000000	4
________Parameter for request number 2		
____________Names		
____________________Tag	0x05	1
____________________Value	0x00000000	4

________Parameter for request number 3		
___________Names		
_________________Tag	0x05	1
_________________Value	0x00000000	4
____signature-length	0x40	1
____signature-content	0x59045AB0F554622B0360340B9D87B5A23ED5723B41DE3F206E58CDD10F915B9FE2E12E2DA3632478A8DF678E418895869AC1E55318CCE04D120D2D6B44DC167B	64
____mac-content	0x5D832D15B57A56D620F198B3	12

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	0x123456789ABCDEF0	8
____date-time		
________length	0x00	1
____other-information		
________Length	0x02	1
________Message Code	0x00B3	2
____content		
________length	0x12	1
access-response		
____tag	0xDA	1
____long-invoke-id-and-priority		
________configuration	0x20	1
________invoke-id	0x000001	3
____date-time	0x00	1
____access-request-specification	0x00	1
____access-response-list-of-data		
____SEQUENCE OF	0x03	1
________Response for request number 1		
________________Tag	0x00	1
________Response for request number 2		
________________Tag	0x00	1
________Response for request number 3		
________________Tag	0x00	1
____access-response-specification		
____SEQUENCE OF	0x03	1
________Result for request number 1		
____________access-response-action	0x03	1
____________result	0x00	1
________Result for request number 2		

_____access-response-action	0x03	1
_____result	0x00	1
_____Result for request number 3		
_____access-response-action	0x03	1
_____result	0x00	1
____signature-length	0x40	1
____signature-content	0xA01AB69ED8A6566AB11643C7358260A88A8B60978593E76A4E93193585D98D9BAD843878F22E794B53F6F280F9F1C848D9D38FC050CDDD58827563E2B0FA19E2	64

ECS12 Set Change of Tenancy date on ESME

Command Message Structure

Name	Encoded Content	Encoded Length
MAC Header (general-ciphering)		
____tag	0xDD	1
____contents	0x000000000000	6
____ciphered-service		
_____length	0x54	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)	0x01	1
_____value (Originator Counter)	0x0000000000000002	8
____originator-system-title		
_____length	0x08	1
_____value	0x123456789ABCDEF0	8
____recipient-system-title		
_____length	0x08	1
_____value	0xFFFFFFFFFFFFFFFE	8
____date-time		
_____length	0x00	1
____other-information		
_____Length	0x02	1
_____Message Code	0x0022	2
____content		
_____length	0x20	1
access-request		
____tag	0xD9	1
____long-invoke-id-and-priority		
_____configuration	0x20	1
_____invoke-id	0x000002	3
____date-time	0x00	1
____access-request-body		
____access-request-specification		
____SEQUENCE OF	0x01	1
_____Request number 1		
_____access-request-set	0x02	1
_____cosem-attribute-descriptor		

_____class-id	0x0001	2
_____instance-id	0x00005E2C0302	6
_____attribute-id	0x02	1
____access-request-list-of-data		
____SEQUENCE OF	0x01	1
_____Parameter for request number 1		
_____Names		
_____Tag	0x09	1
_____Length	0x0C	1
_____Value	0x07DF0105FF000000008000FF	12
____signature-length	0x00	1
____mac-content	0x0F1DD00D6745EBD8E0A663A4	12

Response Message Structure

Name	Encoded Content	Encoded Length
MAC Header (general-ciphering)		
____tag	0xDD	1
____contents	0x000000000000	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	0x000002	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	0x0B3C1B312CEAE9C130060E29	12

7578

19 Use Cases

The Use Cases are contained in the embedded HTML document at Table 19.3. Each Use Case represents one or more interactions with a Device that make up a GBCS Command, Response and / or Alert. This Section 19 provides an overview of the repeatable content within these Use Cases.

By definition, logs can contain zero entries (e.g. because the Device is newly installed or, where allowed by SMETS/CHTS, a log has just been cleared). Any lower size limits for logs in Use Cases, Message Templates and the Mapping Table shall be interpreted accordingly.

19.1 Use Case Title

Each Use Case Title section in Table 19.3 provides common information regarding the Use 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's 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 Service Reference invoked This table is sorted alphabetically by the entry in the column 'name' concatenated with the entry in the column 'attribute / method'

Table 19.1: SMETS / CHTS content of Use Cases

19.1.1 Use Case Cross Reference Section

Table 19.3 provides an overview of important information relevant to the Use Case. It has a 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

Cross Reference	Possible Values	Notes
Requires protection against replay?	Yes / No	Needed to identify which GBCS requirements apply. See Section 4.3.1.5
Service Reference	[e.g. 6.20 SetDeviceConfiguration(MPxN)]	Traceability to other Smart Energy Code documents
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) Access Control Broker (C) Load Controller (C) Load Controller (NC)	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

Table 19.1.1: Allowable values for SMETS / CHTS Use Case Cross References

19.1.2 Objects Applicable to Use Case Section

This section in Table 19.3 contains a 'SMETS objects applicable to Use Case' table to provide traceability between SMETS functions and methods and the Use Case.

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

Table 19.1.2: SMETS objects applicable to Use Case

Beneath that 'SMETS objects applicable to Use Case' table, there is, in each Use Case, one or two sets of detail relating to either (1) the Use Case's DLMS COSEM implementation or (2) the Use Case's GBZ implementation or (3) both.

Each such section begins with a statement as to:

- the 'Target' Device Types that must support the Use Case. Where the 'Target' includes ESME, an SAPC may or shall support the Use Case, as determined by whether the 'SAPC support' column in the 'Use Case reference' tab of Table 20 states 'optional' or 'mandatory' respectively; and
- the required Message Code.

19.1.3 Pre-conditions

Pre-conditions represent conditions for which Device logic is required to ensure correct operation of commands contained within a message, on the Device. Exception conditions (such as failures) that are managed by the Protocol are not captured as Pre-conditions. Manufacturers of Devices must only enforce Pre-conditions that are stated in the Use Cases. Note that the use of Pre-conditions is minimised in favour of controls being implemented on Service User systems.

19.1.4 Actions

Actions stipulate additional Device actions that must be performed together with successful 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 DLMS 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)ttribute 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

19.3 Embedded Use Cases

Table 19.3 contains the Use Cases that fulfil the interface requirements to cover Commands (and their Responses) and Alerts (where applicable). In addition, it includes ZSE Message Templates.

Note: DLMS COSEM methods that have values which have an impact on the execution of the method (that is, methods with input values that are not integer(0)), the DLMS part of the Mapping Table and the Use Case include two or more rows. One row contains the method, and the subsequent row(s) contain the value(s) to be sent with the method.

A number of Use Cases are also covered in GBCS main body. These are identifiable from the Table of Contents.

SECMP0007-DP164-Markup___19-3___Use

GBCS v4.1 Use Cases.html

Field Code Changed

Table 19.3: Use Cases

20 Mapping Table

Table 20 contains the Mapping Table from which the Use Cases and Message Templates were generated. These tables map between SMETS attributes and methods, SEC Service Requests, Use Cases, DLMS COSEM attributes and methods and ZSE clusters, attributes and commands.

In addition to the Use Cases, certain columns in the Mapping Table are directly referenced from this document.

Please note that in the SMETS required objects tab only rows marked ‘E’ (External to HAN) in column F are fully specified, since those rows relate to Remote Party Messages. Other rows are only specified to the extent that these elements of Remote Party Messages rely on them.

SECMP0007-DP164-Markup___Table_20__ GBCS v4.1 Table 20.xlsx

Field Code Changed

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

A Command described in SMETS.

Active Energy

The integral with respect to time of the Active Power in units of watt-hours (Wh) or standard multiples thereof (for example, kWh).

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.

Authenticated Decryption

Has the same meaning as specified in *NIST Special Publication 800-38D*: http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf

Authenticated Encryption (AE)

Has the same meaning as specified in *NIST Special Publication 800-38D*: http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf

Authentication

The method used to confirm the identity of entities or Devices wishing to communicate and 'Authenticated' and 'Authenticity' shall be construed accordingly.

Authentication Key

Shall be as defined in the Green Book.

Authorisation

The process of granting access to a resource and 'Authorised' shall be construed accordingly.

Authorised Public Key Infrastructure (APKI)

A key infrastructure that is compliant with the Certificate related requirements of this GBCS.

Auxiliary Controller

Shall have the meaning defined in SMETS.

Auxiliary Controller Calendar

Shall have the meaning defined in SMETS.

Auxiliary Load Control Switch

Shall have the meaning defined in SMETS.

Blue Book

The DLMS Blue Book Version *DLMS UA 1000-1 Ed. 12.0*. This document can be obtained from the DLMS User Association: http://www.dlms.com.

Boost Function

ESME and SAPC functionality described in SMETS.

Break On Error

Shall have the Green Book meaning of Break On Error used in relation to 'Processing_Option'.

Business Originator

The Smart Metering Entity sending the first Message in a Use Case.

Business Target

The Smart Metering Entity receiving the first Message in a Use Case.

Certificate

An electronic document that binds an identity, and possibly other information, to a Public Key.

Certification Request

A message requesting the issue of a Certificate by a Certification Authority.

Certification Authority (CA)

A trusted entity which issues Certificates.

Certification Authority Certificate

A Certificate issued to a Certification Authority that allows Certification Path Validation in relation to Remote Party's Certificates.

Certification Path Validation

Shall have the meaning defined in Section 4.3.2.8.

Certification Revocation List (CRL) Validation

Shall have the meaning defined in Section 4.3.2.8.

Ciphered Information

Shall have the meaning defined in Section 8.4.

Ciphertext

An output of the Authenticated Encryption function or an input of the Authenticated Decryption function defined in *NIST Special Publication 800-38D*. The unencrypted form of the Ciphertext is the Plaintext.

Clock

A timing mechanism that has a minimum resolution of 1 second.

Command

An instruction to perform a function received or sent via any interface.

Command Response Alert (CRA) Flag

An element within a Message Header that enumerates whether the Message is a Command or a Response or an Alert.

Commercial Product Assurance Security Characteristic

The security characteristics for the relevant Device as indicated in Section 1.0.

Communications Hub

A device complying with the CHTS.

Communications Hub Function (CHF)

A Communications Hub Function as defined in the Communications Hub Technical Specifications.

Communications Hub Technical Specifications (CHTS)

Communications Hub Technical Specifications set out in Schedule 10 of the Smart Energy Code.

Confidentiality

The state of information, in transit or at rest, where there is assurance that it is not accessible by Unauthorised parties through either unintentional means or otherwise.

Consumer

A person who lawfully resides at the Premises that is being Supplied.

Consumer Access Device (CAD)

A Device which, in terms of this GBCS, is permitted to use the same Messages as an IHD.

Consumption

In the context of GSME, Gas Consumption or in the context of ESME, Electricity Consumption information.

Contingency Key

A feature of Trust Anchor Management Protocol (RFC 5934), and only ever used in a recovery scenario when the `root` Certificate (Apex Trust Anchor) needs to be replaced.

Critical Message

A Remote Party Message which may relate to supply being affected, financial fraud or the compromise of Device security. Critical, Critical Commands, Critical Alerts and Critical Responses shall be construed accordingly.

Cryptographic Algorithm

An algorithm for performing one or more cryptographic functions which may include Encryption; Decryption; Digitally Signing or Hashing of information, data, or messages; or exchange of Security Credentials.

Cryptographic Protection

A part of a Message constructed to provide assurance to the Message recipient in terms of one or more of integrity, authenticity, non-repudiation and Confidentiality.

Currency Units

The units of monetary value in major and minor units.

Current Private Key

A Device's Private Key for which the Device has most recently successfully received and processed a Certificate for the corresponding Public Key as defined in Section 13.5.

Data and Communications Company (DCC)

The holder of the licence for the provision of a smart meter communication service granted pursuant to section 6(1)(f) or 6(1A) of the Electricity Act 1989 or section 7AB of the Gas Act 1986.

Data Store

An area of a Device capable of storing information for future retrieval.

Decryption

The process of converting Encrypted information by an Authorised party to recover the original information. Like terms shall be construed accordingly.

Device

A Device that is one of ESME, GSME, Gas Proxy Function, Communications Hub Function, HCALCS, PPMID, SAPC or a Type 2 Device.

Device Based Access Control (DBAC)

Shall have the meaning defined in Section 13.7.3.

Device Certificate

Shall have the meaning set out in Section 12.

Device Log

A data item having the meaning set out in SMETS or CHTS (as relevant).

Digital Signature

The information appended to a Message which is created using the sender's Private Key, that can be verified using the corresponding Public Key contained in the sender's Certificate, and provides the receiver with assurance that the sender is who they claim to be, the message has not been altered in transit and that the holder of the the sender's Private Key created the Message.

Digital Signing

The creation of a Digital Signature.

Digital Signing Certificate

A Certificate which states that the Public Key contained within, and its associated Private Key, may be used for Digital Signing purposes.

Distinguished Encoding Rules

Shall have the meaning defined in https://www.itu.int/rec/T-REC-X.690/en

DLMS COSEM

Device Language Message Specification / Companion Specification for Energy Metering - an Application Layer protocol.

Dual Band Communications Hub (Dual Band CH)

A Communications Hub that is capable of operating at 2.4 GHz and Sub GHz frequencies simultaneously.

Electricity Consumption

The Active Energy Imported into the Premises and 'Consumed' shall be construed accordingly.

Elliptic Curve DSA (ECDSA)

The Elliptic Curve Digital Signature Algorithm (see http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf) as specified in Section 4.3.3.

Encoding(X)

The encoding of a variable length integer X, as specified in Section 3.3.

Encryption

The process of converting information in order to make it unintelligible other than to Authorised parties. Like terms shall be construed accordingly.

Encryption Originator Counter

Shall have the meaning defined in Table 23.

Encryption Remote Party

The Remote Party that can decrypt Encrypted data items.

Entity Identifier

A 64 bit unsigned integer uniquely identifying a Smart Metering Entity.

ESME

Electricity Smart Metering Equipment, as described in SMETS.

ESMETS

The Electricty Smart Metering Equipment (ESME) Technical Specifications in SMETS.

Event

A change in state generated by a Device in response to an internal or external trigger.

Event Code

A 16 bit unsigned integer taking the values specified in Section 16, which equates to the ZigBee *Event ID* parameter. The Alert Code and Event Code are the same for a given Event.

Event Log

Data item described in the Technical Specifications.

Execution Counter

Shall have the meaning defined in Section 4.3.1.5 and in Section 13.3.5.3.

Firmware

The embedded software programmes and / or data structures that control Devices.

Firmware Distribution Receipt Alert

Shall have the meaning set out in Section 11.2.6.

Force Replace

The means to instruct a Communications Hub to replace an ESME, SAPC or GSME Firmware image that it holds, e.g. when the image has only been partially downloaded to the ESME, SAPC or GSME. This enables recovery from failures.

Gas Consumption

The volume of gas in cubic metres (m3) Supplied to the Premises and 'Consumed' shall be construed accordingly.

Gas Proxy Function (GPF)

A Gas Proxy Function as defined in the Communications Hub Technical Specifications.

Galois Counter Mode (GCM)

The mode of operation specified in *NIST Special Publication 800-38D*.

GBZ

A set of structures in the GBCS which carry ZCL / ZSE commands.

General Block Transfer (GBT) / GBT Message

General Block Transfer is a DLMS COSEM mechanism for decomposing APDUs above maximum sizes that can be transported in to a number of smaller APDUs, which are no larger than the maximum sizes. A GBT Message is one of these smaller APDUs.

GMAC

Variant of GCM that is used to generate Message Authentication Code from non-Confidential data, as specified in *NIST Special Publication 800-38D.*

Green Book

The DLMS Green Book Version *DLMS UA 1000-2 Ed.8*. This document can be obtained from the DLMS User Association: http://www.dlms.com.

GSME

Gas Smart Metering Equipment, as described in SMETS.

GSMETS

The Gas Smart Metering Equipment (GSME) Technical Specifications in SMETS.

HAN Only Message

A Message where both the sender and recipient are Devices on the same Smart Metering Home Area Network.

HAN Connected Auxiliary Load Control Switch (HCALCS)

HAN Connected Auxiliary Load Control Switch, as described in SMETS..

Hashing

A repeatable process to create a fixed size condensed representation of a Message or any arbitrary data, as further set out in Section 4.3.3. Hash and like terms shall be construed accordingly.

HCALCSTS

The HAN Connected Auxiliary Load Control Switches (HCALCS) Technical Specifications in SMETS.

Home Area Network Interface (HAN Interface)

A component of GSME, ESME, IHD or other Device that is capable of sending and receiving information to and from other Devices.

IHDTS

The In Home Display Technical Specifications in SMETS.

IHD

In Home Display with its SMETS meaning.

Initialization Vector (IV)

An input to the Authenticated Encryption and Authenticated Decryption functions defined in *NIST Special Publication 800-38D*. Where the GBCS applies, it shall have the values as specified at Section 4.3.3.4.

Inter-PAN

Shall have the meaning defined in CHTS.

Join

The process of authorising two Devices to communicate at the application layer.

Key

Data used to determine the output of a cryptographic operation.

Key Agreement

A means to calculate a shared Key between two parties.

Key Agreement Certificate

A Certificate which states that the Public Key contained within, and its associated Private Key, may be used for Key Agreement purposes.

Key Derivation Function (KDF)

A function to generate derived keying material from a Shared Secret and other information.

Known Remote Party (KRP)

In the context of a specific Device, a Remote Party whose Security Credentials are stored on that Device in at least one Trust Anchor Cell.

KRP Signature

A Digital Signature generated by a Known Remote Party.

Len(X)

The number of octets in the variable length octet string X.

Load Controller

In the context of a specific Device, the Known Remote Party whose Security Credentials are stored in the { loadController, digitalSignature, management} Trust Anchor Cell.

MAC Header

As defined in Section 6, a part of a message which is only present when the Message contains a MAC but which is additional to the MAC.

Manufacturer Image Hash

Shall have the meaning defined in Section 11.2.4.

Mapping Table

The spreadsheet detailing Use Cases and associated protocol requirements as embedded in Section 20.

Maximum Credit Threshold

Shall have the meaning defined in SMETS.

Maximum Meter Balance Threshold

Shall have the meaning defined in SMETS.

Message

A Command, Response or Alert.

Message Authentication

The process by which the receiver of a Message is provided with assurance that the sender is who they claim to be and that the Message is in the form originally sent.

Message Authentication Code (MAC)

The number incorporated in a Message to provide Message Authentication, as set out in Section 4.3.3.

Message Category

A grouping of Remote Party Messages.

Message Code

A 16 bit unsigned integer identifying the Use Case that the Message in question must conform to. Message Codes have the values specified in Section 15.

Message Identifier

Message Identifier shall be the concatenation of:

- Business Originator ID;
- Business Target ID;
- CRA Flag; and
- Originator Counter.

Message Series

Shall have the meaning defined in Section 7.2.11.1.

Message Template

A protocol-specific table defining the encoding of a Message.

Message Type

The Message Types are Command, Response or Alert.

Network Interface

A WAN Interface or HAN Interface.

Network Operator

In the context of a specific Device, the Known Remote Party whose Security Credentials are stored in the {`networkOperator, digitalSignature, management`} Trust Anchor Cell.

Object Identifier (OID)

An identifier used to name an object. Structurally, an OID consists of a node in a hierarchically-assigned namespace, formally defined using the ASN.1 standard.

Operating Sub GHz Channel

The Sub GHz Channel on which a Device is currently operating.

Organisation Certificate

Shall have the meaning set out in Section 12.

Originator Counter

Shall have the meaning defined in Section 4.3.1.2.

OTA Header

Shall have the meaning defined in Section 11.2.3.

OTA Upgrade Image

Shall have the meaning defined in Section 11.2.3.

OtherInfo

An input to the KDF with the meaning as specified in section 5.8.1 of *NIST Special Publication 800-56Ar2*.

Other User

A Remote Party which is not a Known Remote Party in relation to any Device, and so is always an Unknown Remote Party in any communication with a Device.

Outcome

The result of executing a Command, expressed as success or failure.

Payload

Part of the Message that provides the message-specific content.

Payment Mode

The information held on Smart Metering Equipment as described at sections 4 and 5 in SMETS.

Pending Private Key

A Private Key held on a Device for which a Device has not successfully received and processed a Device Certificate for the corresponding Public Key as defined in Section 13.5.

Plain Format

A Signature is a pair of integers, r and s. For the Elliptic Curve required by the GBCS, each can be represented as a 256 bit (or 32 octet) string. The Plain Format of a GBCS signature is the concatenation R || S where R is the 32 octet string representing r and S is the 32 octet string representing s. Thus, a GBCS Signature is an octet string of length 64.

Plaintext

An input to the Authenticated Encryption function and an output from the Authenticated Decryption function defined in *NIST Special Publication 800-38D*. Plaintext is the data whose Confidentiality is to be protected by Encryption. The encrypted form of the Plaintext is the Ciphertext.

PPMIDTS

The Prepayment Interface Device (PPMID) Technical Specifications in SMETS.

Polyphase

ESME containing three measuring elements suitable for a polyphase supply with up to three phases and neutral.

Premise(s)

The premise(s) which is / are being Supplied.

Prepayment Daily Read Log

Shall have the meaning defined in SMETS.

Prepayment Interface Device (PPMID)

A Device that provides a User Interface for Prepayment Mode related information and Commands.

Prepayment Token Decimal (PPTD)

Shall have the meaning defined in Section 14.1.

Prepayment Top Up

The addition of credit to an ESME (or, where it supports it, an SAPC) or GSME operating in prepayment mode.

Prepayment Top Up Token

Shall have the meaning defined in Section 14.1.

Private Digital Signing Key

A Private Key used for Digital Signing only.

Private Key

The key in a Public-Private Key Pair which must be kept secure by the entity to which it relates.

Private Key Cell

Shall have the meaning defined in Section 4.3.2.3. A Private Key Cell may be Current or Pending.

Private Key Agreement Key

A Private Key used for Key Agreement only.

Protection Against Replay

An attribute defined in a Use Case specifying whether a recipient Device is required to implement the Protection Against Replay mechanisms, as defined in Section 4.3.1.5, for the Command covered by the Use Case.

Protocol Data Unit (PDU)

Information delivered as a unit among peer entities of networks containing control information, address information or data.

Provide Security Credentials Command

A Command of a type constructed according to either Section 13.2 or Section 13.9

Public Digital Signing Key

A Public Key used for Digital Signing only.

Public Key

The key in a Public-Private Key Pair which can be distributed to other parties.

Public Key Agreement Key

A Public Key used for Key Agreement only.

Public Key Security Credentials

Security Credentials which include a Public Key.

Public-Private Key Pair

Two mathematically related numbers that are used in Cryptographic Algorithms.

Recovery

In the context of a specific Device, the Known Remote Party whose Security Credentials are stored in the {`recovery, digitalSignature, management`} Trust Anchor Cell.

Reliable Time

The state of the Device clock such that is within 10 seconds of UTC, synchronised with the HAN time server and confirmed by Set Clock Command from the Remote Party whose security Credentials are stored in the {`supplier, digitalSignature, management`} Trust Anchor Cell.

Remote Party

An entity which is remote from a Device and is able to either send Messages to or receive Messages from a Device, whether directly or via a third party.

Remote Party Alert

Shall have the meaning defined in Section 7.2.3.

Remote Party Command

Shall have the meaning defined in Section 7.2.1.

Remote Party Message

A Message where either the sender(s) or recipient(s) are not Devices.

Remote Party Role

The role of a Remote Party in relation to which one or more Devices is capable of storing Security Credentials.

Replay Attack

A form of attack on a Communications Link in which a valid information transmission is repeated through interception and retransmission.

Response

A response to a Command received or sent over any interface.

Response Payload

The parts of a Response that are not related to Cryptographic Protections for integrity, authenticity or non-repudiation, as defined in Section 7.2.2.

Role

The entitlement of a party to execute one or more Commands.

Root

In the context of a specific Device, the entity whose Security Credentials are stored in the {`root, keyCertSign, management`} Trust Anchor Cell.

SAPC

Standalone Auxiliary Proportional Controller, as described in SMETS.

SAPCTS

The Standalone Auxiliary Proportional Controller (SAPC) Technical Specifications in SMETS.

Security Credential Document

Either a:

- Device's Certificate; or a
- Remote Party's Certificate; or a
- Certification Authority Certificate

Security Credentials

Information used to Authenticate a Device, Party or system.

Security Log

Data item described in the Technical Specifications.

Service Reference

Shall have the meaning as set out in Appendix AD of the Smart Energy Code.

SHA-256

The Hashing algorithm of that name approved by the NIST (see http://csrc.nist.gov/groups/ST/toolkit/secure_hashing.html).

Shared Secret

A number which is established by two parties through the Key Agreement technique specified in this GBCS and which can be used as input to a KDF.

Shared Secret Key

A number which is derived using the KDF specified in this GBCS.

Smart Energy Code (SEC)

The document of that name, as designated by the Secretary of State under Condition 22 of the DCC Licence.

Smart Metering Device to Known Remote Party MAC (SMD-KRP MAC)

A MAC generated by a Device in relation to a Response or Alert which can only be verified by the Known Remote Party which is the target of the Response or Alert.

Smart Metering Entity

An entity that is either a Device or a Remote Party.

Smart Metering Equipment Technical Specifications (SMETS)

A version of the Smart Metering Equipment Technical Specifications set out in Schedule 9 of the Smart Energy Code, including the ESMETS, GMSETS, HCALCSTS, PPMIDTS and SAPCTS.

Smart Metering Home Area Network (SMHAN)

The network enabling communications between the Devices recorded within a Communications Hub Function's Device Log (as defined in CHTS).

SMD Signature

A Digital Signature generated by a Device.

Sub GHz

The range of radio frequencies catered for by the ZSE 'Sub-GHz Cluster'.

Sub GHz Alert

An Alert which, in the column headed '0x02 CHF' in Table 16.2, is required to be issued by a 'Dual Band CH Only'. All such Alerts are generated by the CHF.

Sub GHz Channel

One of the radio frequencies catered for by the ZSE 'Sub-GHz Cluster'.

Sub GHz Channel Masks

Configuration settings to limit the Sub GHz frequencies that a Dual Band CH can switch to.

Sub GHz Available Channels

The subset of Sub GHz frequencies that could be used by Dual Band CH and so could be set in the Sub GHz Channel Masks, so as to comply with UK telecommunications regulations. This subset of Sub GHz frequencies is defined in Section 10.6.5.

Sub GHz Channel Change

The Dual Band CH action of changing its Operating Sub GHz Channel.

Sub GHz Channel Scan

A process undertaken by a Dual Band Communications Hub to identify the level of radio noise on each of the Sub GHz Channels that it is allowed to use by the Sub GHz Channel

Masks. This is also referred to in the IEEE 802.15.4a 2007 standard as 'active channel scan' and in ZSE as 'Energy Scan'.

Sub GHz End Device

A Device which is capable of operating on Sub GHz frequencies and which is not a Communications Hub.

Sub GHz GSME

A GSME that is capable of operating on Sub GHz frequencies.

Sub GHz Non-GSME Device

A Sub GHz End Device that is not a GSME.

Subject Unique ID

In the context of a Certificate, shall mean the value field of the `AttributeTypeAndValue` structure within the `subject` field whose type is `id-at-uniqueIdentifier`.

Supplementary Originator Counter

Shall have the meaning defined in Section 4.3.1.4.

Supplementary Remote Party

The Remote Party identified by a Supplementary Remote Party ID.

Supplementary Remote Party ID

In relation to an Alert which has two recipient roles identified as required, the Entity Identifier of the recipient which is not the Supplier shall be the Supplementary Remote Party ID. In relation to a Command and the resulting Response, the Supplementary Remote Party ID shall be the Entity Identifier of the Remote Party requesting the creation of the Command by the Access Control Broker.

Supplementary Remote Party Counter

In relation to a Command and the resulting Response, a 64 bit integer specified by the Remote Party requesting the creation of the Command by the Access Control Broker.

Supply

The supply of gas to Premises for GSME and the supply of electricity to Premises for ESME and 'Supplied' shall be construed accordingly.

Supplier

A person authorised by licence to Supply gas to Premises for GSME and a person authorised by licence to Supply electricity to Premises for ESME. In the context of a specific Device, the Known Remote Party whose Security Credentials are stored in the {`supplier, digitalSignature, management`} Trust Anchor Cell.

Tag

The first element within a Message Header or part of a Message that provides identification of the Message or part of Message that follows.

Tapping Off Mechanism (TOM)

Shall have the meaning defined in Section 10.3.4.

Tariff

The structure of prices and other charges relating to a Supply.

Tariff Block Counter Matrix

Data item described in SMETS.

Technical Specifications

The document set comprising SMETS (incorporating the ESMETS, GSMETS, IHDTS, HCALCSTS, PPMIDTS and SAPCTS), and CHTS.

TOU

Time of Use.

Transitional Change of Supplier

In the context of a specific Device, the Known Remote Party whose Security Credentials are stored in the `{transitionalCoS, digitalSignature, management}` Trust Anchor Cell.

Trust Anchor (TA)

A Trust Anchor represents a Remote Party via a Public Key and associated data stored on a Device. A Trust Anchor is used by the Device in specified cryptographic operations to determine whether it should act on Remote Party Commands received.

Trust Anchor Cell

A data store on a Device capable of storing one Trust Anchor. Each Trust Anchor Cell is for a fixed and pre-specified `KeyUsage`, `CellUsage` and `RemotePartyRole`.

Trust Anchor Management Protocol (TAMP)

A range of IETF RFCs relate to Trust Anchor Management, including:

- [RFC4210] Adams, C., Farrell, S., Kause, T., and T. Mononen, 'Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)', RFC 4210, September 2005.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, 'Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile', RFC 5280, May 2008.
- [RFC5914] Housley, R., Ashmore, S., and C. Wallace, 'Trust Anchor Format', RFC 5914, June 2010.
- [RFC5934] Housley, R., Ashmore, S., and C. Wallace, 'Trust Anchor Management Protocol (TAMP)', RFC 5934, August 2010.
- [RFC6024] Reddy, R. and C. Wallace, 'Trust Anchor Management Requirements', RFC 6024, October 2010.

Twin Element

ESME containing two measuring elements.

Type 2 Device

A Device that is not required to have a Device Log, with its SMETS meaning.

Unauthorised

Not Authorised.

Unique Transaction Reference Number (UTRN)

A 20 decimal digit number that is used to convey a Prepayment Top-Up Remote Party Command to an ESME / GSME / SAPC (where it supports such functionality).

Unknown Remote Party (URP)

In the context of a specific Device, a Remote Party whose Security Credentials are not stored on that Device.

Update Security Credentials Command

A Command of a type constructed according to either Section 13.3 or Section 13.10.

Upgrade Image

Shall have the meaning defined in Section 11.2.2.

Use Case

The structure, format and processing of a Message.

User

Shall have the meaning as set out in Section A of the Smart Energy Code

User Interface

An interface for providing local human interaction with Devices which supports input and visual output.

User Interface Command

A Remote Party Command that is entered through the User Interface.

UTC

Coordinated Universal Time.

UTRN Check Digit

Shall have the meaning defined in Section 14.1.

UTRN Counter Cache

Shall have the meaning defined in Section 14.1.

Variant Message

A Message that does not fall in to any of the Message Categories defined in Section 6.

Wide Area Network (WAN) Interface

A component of a Communications Hub that is capable of sending and receiving information via the Wide Area Network Provider.

Wide Area Network (WAN) Provider

The organisation providing communications over the WAN Interface of the Communications Hub. Consequently, in the context of a specific Communications Hub, the Known Remote Party whose Security Credentials are stored in the {`wanProvider`, `digitalSignature`, `management`} Trust Anchor Cell.

ZigBee Cluster Library (ZCL)

The ZigBee Cluster Library Specification reference document as defined in the 'Documentation Alignment' Section of this GBCS.

ZigBee PRO/2007 Layer PICS and Stack Profiles

The ZigBee PRO/2007 Layer PICS and Stack Profiles reference document as defined in the 'Documentation Alignment' Section of this GBCS.

ZigBee Specification

The ZigBee Specification reference document as defined in the 'Documentation Alignment' Section of this GBCS.

ZigBee Smart Energy (ZSE) Profile (SEP)

The ZigBee Smart Energy (ZSE) Profile Specification reference document as defined in the 'Documentation Alignment' Section of this GBCS.

22 Annex 1 – Additional DLMS Class

The class described below shall be supported by ESME, and shall be supported by SAPC where SAPC supports Use Cases that require this class.

Extended Data (class_id: 9000 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			

22.1 Attribute description

logical_name	Identifies the 'Data' object instance	

value_active	Contains the data.
	CHOICE
	{
	-- 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]
	}
		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.

scaler_unit_active	Provides information on the unit and the scalar for the value. scal_unit_type: structure	

	{
	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

value_passive	Contains the data.
	CHOICE
	{
	-- 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]
	}
	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.

scaler_unit_passive	Provides information on the unit and the scalar for the value.
	scal_unit_type: structure
	{
	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

activate_passive_value_time	Defines the time when the object itself calls the specific method activate_passive_value. 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. octet-string, formatted as set in 4.1.6.1 for date_time of the Blue DLMS Book

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)

23 Annex 2 – This Section is intentionally left blank

24 Annex 3 – This Section is intentionally left blank

25 Annex 4 – Use of ZigBee in GBCS – informative

25.1 Purpose

This annex briefly summarises where the GBCS:

- requires the use of ZigBee, specifically where it uses parts of the ZigBee specifications, or takes an approach which aligns to the ZigBee specifications; and
- does not allow the use of ZigBee / requires its use to be modified, specifically where it:
  - mandates a solution that is not ZigBee derived but where there is ZigBee equivalent in the specifications;
  - specifies an approach that is derived from ZigBee but the approach is not part of the ZigBee specifications; and
  - specifies an approach that uses parts of the ZigBee but varies from it on specific points.

The document is based on the content of ZigBee specifications referenced in the GBCS.

25.2 GBCS requirements to use ZigBee

For all Smart Metering Equipment, the GBCS requires the implementation of functionality equivalent to a subset of the ZigBee standards, including all mandatory components required to achieve ZSE certification.

GBCS and the ZigBee standards specify all items that need to be certified.

25.3 GBCS requirements not to use ZigBee / vary from it

This Section summarises areas where GBCS requires that parts of the ZigBee standards are not used or used in a different way. It is an informative summary. The normative requirements are as stated in the normative Sections of this document.

For some communications between Devices, GBCS requires a solution other than solely ZigBee. For example, Commands from a PPMID to a GSME (and the corresponding Responses) are GBCS specified and carried in ZSE TransferData commands.

GBCS requires certain ZigBee features to have a GB-specific interpretation. For example, Section 10.4.2.11 specifies how the ZSE AccumulatedDebt attribute shall be populated by the GSME and interpreted by all other devices.

GBCS requires specific population of certain ZigBee commands for GB usage. For example, Section 10.4.2.11 specifies how the ZSE Get Event Log and Clear Event Log commands can be used to access GSME Proxy Log copies of the GSME Event or Security Log on the GPF.

GBCS requires a different interpretation of certain ZigBee parameters to support GB usage. For example, Section 10.4.2.11 specifies that when using the ZSE GetSampledData and GetSampledDataResponse commands, a SampleRequestInterval field shall contain 0xFFFF whenever the SampleID field is 0x0001. GBCS specifies that a SampleID of 0x0001 corresponds to the Daily Consumption Log, and SMETS specifies that this log contains reading data taken at midnight UTC, so once every 24 hours. This is a longer period than can be specified in the SampleRequestInterval, given that parameter's size and hence the deviation from the standard.

GBCS lays out specific requirements regarding the use of Inter-PAN communications between a Communications Hub and an HHT. These are stated in Section 10.

GBCS bars or modifies some internal Device behaviour that is specified in ZigBee standards, for example:

- The ZSE capability for the OTA Client to self-activate any Firmware is not allowed, as stated in Section 11.2.1.
- The ZSE constraint requiring Trust Center involvement in CBKE shall not be applied, as stated in Section 13.7.2.

GBCS does not require ZSE certification either for non-standard ZSE features or for non-ZSE features in GBCS.

26 Annex 5 – Use of DLMS COSEM in GBCS – informative

Statements in the Section regarding an ESME apply to an SAPC as if it were an ESME, where the SAPC supports the functionality to which the statement relates.

Statements that do not apply to an ESME also do not apply to an SAPC.

26.1 Purpose

This annex briefly summarises where the GBCS:

- requires the use of DLMS COSEM: specifically where it uses parts of the DLMS COSEM specification, or takes an approach which aligns to the DLMS COSEM specification; and
- does not allow the use of DLMS COSEM / requires its use to be modified: specifically where it:
  - Mandates a solution that is not DLMS COSEM derived but where there are DLMS COSEM equivalents;
  - Specifies an approach that is derived from DLMS COSEM but the approach is not part of the DLMS COSEM specification; or
  - Specifies an approach that uses parts of the DLMS COSEM but varies from it on specific points.

26.2 GBCS requirements to use DLMS COSEM

For ESME and CHF, the GBCS requires the implementation of functionality equivalent to a subset of the Blue Book Classes. It does not require functionality equivalent to other Blue Book classes not identified in the GBCS.

For all Devices, GBCS requires a set of cryptographic primitives that align to DLMS Security Suite 1, and so all Devices will need functionality which is in line with the cryptography related parts of the Green Book (for both GBCS and DLMS COSEM, these requirements are NIST derived).

GBCS requires that all Devices use X.509 Certificates and Certification Requests with a number of optional elements being used / barred. These requirements align with the Green Book requirements (which are X.509 derived).

For ESME and CHF, the GBCS requires functionality equivalent to Green Book access and data notification services.

For all Devices, the GBCS requires functionality equivalent to the Green Book's general ciphering and general signing services.

For all Devices, the GBCS requires functionality equivalent to the Green Book's authenticated encryption and decryption.

For all Devices, the GBCS requires corresponding alignment with DLMS COSEM's ASN.1 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 GBZ / 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 GBZ 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 ZigBee 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, not specified in the Blue Book.

For large Remote Party Messages, the GBCS uses DLMS type structures but in a way not specified in the DLMS COSEM specification.

Messages using pairwise key agreement between GSME and PPMID uses a structure similar to DLMS COSEM's message structure, but that structure 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 values 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).

27 Annex 6 – Deducing the UTRN Counter from the 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.

In this case, the UTRN Counter being entered into the Device is 5 greater than the highest thus far received by the Device.

Parameter	Value (Binary)	Decimal Representation
Vended by supplier		
Originator Counter (64 bits)	1001001010001111110001110010110000000000000000000000000000000000	10,560,878,642,999,590,912
UTRN Counter (32 bits)	10010010100011111100011100101100	2,458,896,172
PTUT Truncated UTRN Counter (10 bits)	1100101100	812
Recorded on Device		
Highest entry in UTRN Counter Cache (32 bits) = *V*	10010010100011111100011100100111	2,458,896,167

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	1001001010001111110001000000000000	2,458,895,360
4	Set *r* = PTUT Truncated Originator Counter	1100101100	812

Step	Description	Example	
		Binary Representation	Decimal Representation
5	Calculate p – 2^9 (Call this variable, *x*) (See footnote 47) x = 807 – 512	100100111	295
6	Calculate p + 2^9 (Call this variable, *y*) y = 807 + 512	10100100111	1319
7	Test r against x and y and set s accordingly • If r < x then s = r + 2^{10} • If r > y then s = r – 2^{10} • Else s = r 295 < 812 < 1319, therefore s = r	1100101100	812
8	Set deduced Originator Counter = (q + s) *2^{32}	1001001010001111110001110010110000000000000000000000000000000000	10,560,878,642,999,590,912
9	Set deduced UTRN Counter as most significant 32 bits of Deduced Originator Counter	10010010100011111100011100101100	2,458,896,172

Table 27: Derivation of the UTRN Counter from the PTUT Truncated UTRN Counter – a worked example

[47] 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.

28 Annex 7 – Data Item Values to be set prior to installation of Devices

Tables 28a and 28b lists data items and values that shall be configured in ESME prior to installation.

ESME shall also be configured prior to installation to give effect to the settings in Tables 28c and 28d.

The values in Tables 28a and 28b, that relate to 'Randomised Offset Limit', shall be configured in SAPC prior to installation.

Where an SAPC supports the functionality associated with a row in Tables 28a, 28b, 28c or 28d, the SAPC shall be configured according to those rows prior to installation.

Device	Data Item	Reference	Value	Notes
ESME (all variants)	Maximum Meter Balance Threshold	SMETS 5.7.4.27	300,000,000 millipence	NA
ESME (all variants)	Randomised Offset Limit	SMETS 5.7.4.33	600 seconds	The Randomised Offset Is the product of the Randomised Offset Limit(5.7.4.33) and the Randomised Offset Number(5.7.1.5) rounded to the nearest second. This value is used to delay the Tariff Switching Table times and the Auxiliary Controller switching times.
ESME (all variants)	RMS Extreme Over Voltage Threshold	SMETS 5.7.4.35	265.0 volts	GBCS Use Cases specify a resolution to tenths of volts
ESME (all variants)	RMS Extreme Over Voltage Measurement Period	SMETS 5.7.4.34	180 seconds	NA
ESME (all variants)	RMS Extreme Under Voltage Threshold	SMETS 5.7.4.37	190.0 volts	NA
ESME (all variants)	RMS Extreme Under Voltage Measurement Period	SMETS 5.7.4.36	180 seconds	NA
ESME (all variants)	RMS Voltage Sag Threshold	SMETS 5.7.4.40	190.0 volts	NA
ESME (all variants)	RMS Voltage Sag Measurement Period	SMETS 5.7.4.38	180 seconds	NA
ESME (all variants)	RMS Voltage Swell Threshold	SMETS 5.7.4.41	265.0 volts	NA
ESME (all variants)	RMS Voltage Swell Measurement Period	SMETS 5.7.4.39	180 seconds	NA

Device	Data Item	Reference	Value	Notes
ESME (all variants)	(Phase[1]) Average RMS Voltage Measurement Period	SMETS 5.7.4.6 (5.19.1.3)	1800 seconds	NA
ESME (all variants)	(Phase[1]) Average RMS Under Voltage Threshold	SMETS 5.7.4.5 (5.19.1.2)	212.0 volts	NA
ESME (all variants)	(Phase[1]) Average RMS Over Voltage Threshold	SMETS 5.7.4.4 (5.19.1.1)	258.0 volts	NA
Polyphase ESME	Phase[2] Average RMS Voltage Measurement Period	SMETS 5.19.1.3	1800 seconds	NA
Polyphase ESME	Phase[2] Average RMS Under Voltage Threshold	SMETS 5.19.1.2	212.0 volts	NA
Polyphase ESME	Phase[2] Average RMS Over Voltage Threshold	SMETS 5.19.1.1	258.0 volts	NA
Polyphase ESME	Phase[3] Average RMS Voltage Measurement Period	SMETS 5.19.1.3	1800 seconds	NA
Polyphase ESME	Phase[3] Average RMS Under Voltage Threshold	SMETS 5.19.1.2	212.0 volts	NA
Polyphase ESME	Phase[3] Average RMS Over Voltage Threshold	SMETS 5.19.1.1	258.0 volts	NA
ESME (all variants)	Maximum Demand Configurable Time Period: - start time	SMETS 5.7.4.26	16:00 in hh:mm	
ESME (all variants)	Maximum Demand Configurable Time Period:- end time	SMETS 5.7.4.26	20:00 in hh:mm	

Table 28a: Data items and values to be configured prior to installation of Devices

Data Item	Reference	COSEM class ID	OBIS Code	Attribute ID	Attribute Name	COSEM datatype	Encoded value	Decoded value
Maximum Meter Balance Threshold	SMETS 5.7.4.27	9000	0-0:94.44.2.20	4	value_ passive	double-long	0x11E1 A300	300,000 ,000
Randomised Offset Limit	SMETS 5.7.4.33	1	0-0:94.44.0.1	2	value	long-unsigned	0x0258	600

Data Item	Reference	COSEM class ID	OBIS Code	Attribute ID	Attribute Name	COSEM datatype	Encoded value	Decoded value
RMS Extreme Over Voltage Threshold	SMETS 5.7.4.35	71	0-0:17.0.1.255	4	threshold_normal	double-long-unsigned	0x00000A5A	2650
RMS Extreme Over Voltage Measurement Period	SMETS 5.7.4.34	71	0-0:17.0.1.255	6	min_over_threshold_duration	double-long-unsigned	0x000000B4	180
RMS Extreme Under Voltage Threshold	SMETS 5.7.4.37	71	0-0:17.0.2.255	4	threshold_normal	double-long-unsigned	0x00000076C	1900
RMS Extreme Under Voltage Measurement Period	SMETS 5.7.4.36	71	0-0:17.0.2.255	6	min_over_threshold_duration	double-long-unsigned	0x000000B4	180
RMS Voltage Sag Threshold	SMETS 5.7.4.40	71	0-0:17.0.3.255	4	threshold_normal	double-long-unsigned	0x00000076C	1900
RMS Voltage Sag Measurement Period	SMETS 5.7.4.38	71	0-0:17.0.3.255	6	min_over_threshold_duration	double-long-unsigned	0x000000B4	180
RMS Voltage Swell Threshold	SMETS 5.7.4.41	71	0-0:17.0.4.255	4	threshold_normal	double-long-unsigned	0x00000A5A	2650
RMS Voltage Swell Measurement Period	SMETS 5.7.4.39	71	0-0:17.0.4.255	6	min_over_threshold_duration	double-long-unsigned	0x000000B4	180
(Phase[1]) Average RMS Voltage Measurement Period	SMETS 5.7.4.6 (5.19.1.3)	7	1-0:32.24.0.255	4	capture_period	double-long-unsigned	0x00000708	1800

Data Item	Reference	COSEM class ID	OBIS Code	Attribute ID	Attribute Name	COSEM datatype	Encoded value	Decoded value
(Phase[1]) Average RMS Under Voltage Threshold	SMETS 5.7.4.5 (5.19.1.2)	1	1-0:32.31.0.4	2	value	double-long-unsigned	0x00000848	2120
(Phase[1]) Average RMS Over Voltage Threshold	SMETS 5.7.4.4 (5.19.1.1)	1	1-0:32.35.0.4	2	value	double-long-unsigned	0x00000A14	2580
Phase[2] Average RMS Voltage Measurement Period	SMETS 5.19.1.3	7	1-0:52.24.0.255	4	capture_period	double-long-unsigned	0x00000708	1800
Phase[2] Average RMS Under Voltage Threshold	SMETS 5.19.1.2	1	1-0:52.31.0.4	2	value	double-long-unsigned	0x00000848	2120
Phase[2] Average RMS Over Voltage Threshold	SMETS 5.19.1.1	1	1-0:52.35.0.4	2	value	double-long-unsigned	0x00000A14	2580
Phase[3] Average RMS Voltage Measurement Period	SMETS 5.19.1.3	7	1-0:72.24.0.255	4	capture_period	double-long-unsigned	0x00000708	1800
Phase[3] Average RMS Under Voltage Threshold	SMETS 5.19.1.2	1	1-0:72.31.0.4	2	value	double-long-unsigned	0x00000848	2120
Phase[3] Average RMS Over Voltage Threshold	SMETS 5.19.1.1	1	1-0:72.35.0.4	2	value	double-long-unsigned	0x00000A14	2580

Table 28b: DLMS COSEM values to be configured in Devices at manufacture.

Event / Alert Code	Event / Alert Code Meaning	Default Configuration - Send WAN Alert Y = Send Alert N = Do not send Alert	Default Configuration - Store Alert in Power Event Log Y = Store in log N = Do not store in log
0x8002	Average RMS Voltage above Average RMS Over Voltage Threshold (current value above threshold; previous value below threshold)	Y	Y
0x8003	Average RMS Voltage above Average RMS Over Voltage Threshold on Phase 1 (current value above threshold; previous value below threshold)	Y	Y
0x8004	Average RMS Voltage above Average RMS Over Voltage Threshold on Phase 2 (current value above threshold; previous value below threshold)	Y	Y
0x8005	Average RMS Voltage above Average RMS Over Voltage Threshold on Phase 3 (current value above threshold; previous value below threshold)	Y	Y
0x8006	Average RMS Voltage below Average RMS Under Voltage Threshold (current value below threshold; previous value above threshold)	Y	Y
0x8007	Average RMS Voltage below Average RMS Under Voltage Threshold on Phase 1 (current value below threshold; previous value above threshold)	Y	Y
0x8008	Average RMS Voltage below Average RMS Under Voltage Threshold on Phase 2 (current value below threshold; previous value above threshold)	Y	Y
0x8009	Average RMS Voltage below Average RMS Under Voltage Threshold on Phase 3 (current value below threshold; previous value above threshold)	Y	Y
0x8020	RMS Voltage above Extreme Over Voltage Threshold (voltage rises above for longer than the configurable period)	Y	Y
0x8021	RMS Voltage above Extreme Over Voltage Threshold on Phase 1 (voltage rises above for longer than the configurable period)	Y	Y
0x8022	RMS Voltage above Extreme Over Voltage Threshold on Phase 2 (voltage rises above for longer than the configurable period)	Y	Y
0x8023	RMS Voltage above Extreme Over Voltage Threshold on Phase 3 (voltage rises above for longer than the configurable period)	Y	Y
0x8024	RMS Voltage above Voltage Swell Threshold (voltage rises above for longer than the configurable period)	N	N
0x8025	RMS Voltage above Voltage Swell Threshold on Phase 1 (voltage rises above for longer than the configurable period)	N	N
0x8026	RMS Voltage above Voltage Swell Threshold on Phase 2 (voltage rises above for longer than the configurable period)	N	N
0x8027	RMS Voltage above Voltage Swell Threshold on Phase 3 (voltage rises above for longer than the configurable period)	N	N

Event / Alert Code	Event / Alert Code Meaning	Default Configuration - Send WAN Alert Y = Send Alert N = Do not send Alert	Default Configuration - Store Alert in Power Event Log Y = Store in log N = Do not store in log
0x8028	RMS Voltage below Extreme Under Voltage Threshold (voltage falls below for longer than the configurable period)	Y	Y
0x8029	RMS Voltage below Extreme Under Voltage Threshold on Phase 1 (voltage falls below for longer than the configurable period)	Y	Y
0x802A	RMS Voltage below Extreme Under Voltage Threshold on Phase 2 (voltage falls below for longer than the configurable period)	Y	Y
0x802B	RMS Voltage below Extreme Under Voltage Threshold on Phase 3 (voltage falls below for longer than the configurable period)	Y	Y
0x802C	RMS Voltage below Voltage Sag Threshold (voltage falls below for longer than the configurable period)	N	N
0x802D	RMS Voltage below Voltage Sag Threshold on Phase 1 (voltage falls below for longer than the configurable period)	N	N
0x802E	RMS Voltage below Voltage Sag Threshold on Phase 2 (voltage falls below for longer than the configurable period)	N	N
0x802F	RMS Voltage below Voltage Sag Threshold on Phase 3 (voltage falls below for longer than the configurable period)	N	N
0x8085	Average RMS Voltage below Average RMS Over Voltage Threshold (current value below threshold; previous value above threshold)	Y	Y
0x8086	Average RMS Voltage below Average RMS Over Voltage Threshold on Phase 1 (current value below threshold; previous value above threshold)	Y	Y
0x8087	Average RMS Voltage below Average RMS Over Voltage Threshold on Phase 2 (current value below threshold; previous value above threshold)	Y	Y
0x8088	Average RMS Voltage below Average RMS Over Voltage Threshold on Phase 3 (current value below threshold; previous value above threshold)	Y	Y
0x8089	Average RMS Voltage above Average RMS Under Voltage Threshold (current value above threshold; previous value below threshold)	Y	Y
0x808A	Average RMS Voltage above Average RMS Under Voltage Threshold on Phase 1 (current value above threshold; previous value below threshold)	Y	Y
0x808B	Average RMS Voltage above Average RMS Under Voltage Threshold on Phase 2 (current value above threshold; previous value below threshold)	Y	Y
0x808C	Average RMS Voltage above Average RMS Under Voltage Threshold on Phase 3 (current value above threshold; previous value below threshold)	Y	Y

Event / Alert Code	Event / Alert Code Meaning	Default Configuration - Send WAN Alert Y = Send Alert N = Do not send Alert	Default Configuration - Store Alert in Power Event Log Y = Store in log N = Do not store in log
0x808D	RMS Voltage above Extreme Over Voltage Threshold (voltage returns below for longer than the configurable period)	Y	Y
0x808E	RMS Voltage above Extreme Over Voltage Threshold on Phase 1 (voltage returns below for longer than the configurable period)	Y	Y
0x808F	RMS Voltage above Extreme Over Voltage Threshold on Phase 2 (voltage returns below for longer than the configurable period)	Y	Y
0x8090	RMS Voltage above Extreme Over Voltage Threshold on Phase 3 (voltage returns below for longer than the configurable period)	Y	Y
0x8091	RMS Voltage above Voltage Swell Threshold (voltage returns below for longer than the configurable period)	N	N
0x8092	RMS Voltage above Voltage Swell Threshold on Phase 1 (voltage returns below for longer than the configurable period)	N	N
0x8093	RMS Voltage above Voltage Swell Threshold on Phase 2 (voltage returns below for longer than the configurable period)	N	N
0x8094	RMS Voltage above Voltage Swell Threshold on Phase 3 (voltage returns below for longer than the configurable period)	N	N
0x8095	RMS Voltage below Extreme Under Voltage Threshold (voltage returns above for longer than the configurable period)	Y	Y
0x8096	RMS Voltage below Extreme Under Voltage Threshold on Phase 1 (voltage returns above for longer than the configurable period)	Y	Y
0x8097	RMS Voltage below Extreme Under Voltage Threshold on Phase 2 (voltage returns above for longer than the configurable period)	Y	Y
0x8098	RMS Voltage below Extreme Under Voltage Threshold on Phase 3 (voltage returns above for longer than the configurable period)	Y	Y
0x8099	RMS Voltage below Voltage Sag Threshold (voltage returns above for longer than the configurable period)	N	N
0x809A	RMS Voltage below Voltage Sag Threshold on Phase 1 (voltage returns above for longer than the configurable period)	N	N
0x809B	RMS Voltage below Voltage Sag Threshold on Phase 2 (voltage returns above for longer than the configurable period)	N	N
0x809C	RMS Voltage below Voltage Sag Threshold on Phase 3 (voltage returns above for longer than the configurable period)	N	N

Event / Alert Code	Event / Alert Code Meaning	Default Configuration - Send WAN Alert Y = Send Alert N = Do not send Alert	Default Configuration - Store Alert in Power Event Log Y = Store in log N = Do not store in log
0x8010	Over Current	N	N
0x8011	Over Current L1	N	N
0x8016	Over Current L2	N	N
0x8013	Over Current L3	N	N
0x8014	Power Factor Threshold Below	N	N
0x8015	Power Factor Threshold Ok	N	N

Table 28c: WAN Alert and Power Event Log settings to be configured prior to installation of Devices

Attribute	COSEM datatype	Tag	Length	Value	Meaning
entries:entries[1..2]	array	0x01	0x02		An array with two entries, the first turns on monitoring and the second turns it off
entries:entries[1]. schedule_table_entry. Index	long-unsigned	0x12		0x0001	The first entry which turns monitoring on
entries:entries[1]. schedule_table_entry. enable	boolean	0x03		0x01	True, so the entry always executes
entries:entries[1]. schedule_table_entry. script_logical_name	octet-string(6)	0x09	0x06	0x00000A8064FF	0-0:10.128.100.255 which, as per Table 7.3.8, is the script table controlling monitoring
entries:entries[1]. schedule_table_entry. script_selector	long-unsigned	0x12		0x0001	Meaning start monitoring at the time in this entry
entries:entries[1]. schedule_table_entry. switch_time	octet-string(4)	0x09	0x04	0x10000000	16:00:00:00 - the time at which monitoring is to turn on
entries:entries[1]. schedule_table_entry. validity_window	long-unsigned	0x12		0xFFFF	The script is processed at any time after power failure
entries:entries[1]. schedule_table_entry. exec_weekdays	bit-string(7)	0x04	0x07	0xFE	0xFE = 0b11111110, which means execute this script every day
entries:entries[1]. schedule_table_entry. exec_specdays	bit-string(0)	0x04	0x00		No special day processing

Attribute	COSEM datatype	Tag	Length	Value	Meaning
entries:entries[1]. schedule_table_entry. begin_date	octet-string(5)	0x09	0x05	0x000001FFFFF	From the start of time
entries:entries[1]. schedule_table_entry. end_date	octet-string(5)	0x09	0x05	0xFFFFFFFFFF	For all time
entries:entries[2]. schedule_table_entry. index	long-unsigned	0x12		0x0002	The second entry which turns monitoring off
entries:entries[2]. schedule_table_entry. enable	boolean	0x03		0x01	True, so the entry always executes
entries:entries[2]. schedule_table_entry. script_logical_name	octet-string(6)	0x09	0x06	0x00000A8064FF	0-0:10.128.100.255 which, as per Table 7.3.8, is the script table controlling monitoring
entries:entries[2]. schedule_table_entry. script_selector	long-unsigned	0x12		0x0002	Meaning stop monitoring at the time in this entry
entries:entries[2]. schedule_table_entry. switch_time	octet-string(4)	0x09	0x04	0x14000000	20:00:00:00 - the time at which monitoring is to turn off
entries:entries[2]. schedule_table_entry. validity_window	long-unsigned	0x12		0xFFFF	The script is processed at any time after power failure
entries:entries[2]. schedule_table_entry. exec_weekdays	bit-string(7)	0x04	0x07	0xFE	0xFE = 0b11111110, which means execute this script every day
entries:entries[2]. schedule_table_entry. exec_specdays	bit-string(0)	0x04	0x00		No special day processing
entries:entries[2]. schedule_table_entry. begin_date	octet-string(5)	0x09	0x05	0x000001FFFF	From the start of time
entries:entries[2]. schedule_table_entry. end_date	octet-string(5)	0x09	0x05	0xFFFFFFFFFF	For all time

Table 28d: Tag, length and values to be populated in attribute 2 of OBIS code 0-0:12.0.0.255 (which relates to the SMETS ‘Maximum Demand Configurable Time Period ‘) to be configured prior to installation of ESME.