

Great Britain Companion Specification (GBCS)

Version 1.01

6 November 2017~~8 November 2016~~

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 1.2a v1.0 reference 07-5356r19;
- ZigBee Cluster Library Specification, reference 07-5123r04;
- ZigBee OTA Upgrade Cluster Specification, reference 09-5264r23;
- ZigBee Specification – 05-3474r20; and
- ZigBee Pro PICS and Stack Profiles – 08-0006r05.

These documents are available from <http://zigbee.org/About/GBCSPartner.aspx>.

Table of Contents

1	Introduction – normative	6
2	Structure of the GB Companion Specification (GBCS)	7
2.1	Normative Requirements	7
2.2	Structure of the GB Companion Specification (GBCS) and its relationship to other documents – informative	7
3	Scope and Terminology	9
3.1	Introduction – informative	9
3.2	Scope	9
3.3	Terminology	10
4	Security	13
4.1	Introduction – informative	13
4.2	Cryptographic Protections applying to all Messages	14
4.3	Security for Remote Party Messages	14
5	Remote Party Message construction, protection and verification – informative 27	
5.1	Common Message Structures – informative	27
5.2	Common Encryption and Decryption approach – informative	27
5.3	Message Categories – informative	27
5.4	Common Message Processing steps – informative	28
5.5	Common processing stages and requirements for Devices operated through the DCC – informative	30
6	Message Categories	34
6.1	Introduction – informative	34
6.2	Message Category SME.C	34
6.3	Message Category SME.C.C	38
6.4	Message Category SME.C.NC	40
6.5	Message Category SME.C.PPMID-GSME	42
6.6	Message Category SME.A	43
6.7	Message Category SME.A.C	44
6.8	Message Category SME.A.NC	45
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	<u>6665</u>
7.4	Device requirements – ZSE	<u>7473</u>
8	Encryption of Attributes in Remote Party Messages	<u>7775</u>
8.1	Approach – informative	<u>7775</u>
8.2	Common requirements	<u>7775</u>
8.3	Key Derivation Inputs	<u>7876</u>
8.4	AAD, Plaintext and Ciphertext	<u>7876</u>
8.5	Access to sensitive data – COSEM attribute access	<u>7977</u>
8.6	Access to sensitive data – ZSE attribute access	<u>8683</u>
9	Time Synchronisation and Future Dated Remote Party Messages	<u>8784</u>
9.1	Time synchronisation	<u>8784</u>
9.2	Future Dated Remote Party Messages	<u>9491</u>

10	ZSE Implementation	<u>9996</u>
10.1	Introduction – informative	<u>9996</u>
10.2	Tunnels	<u>9996</u>
10.3	GSME and GPF interactions	<u>104104</u>
10.4	GPF Structured Data Items	<u>107104</u>
10.5	Hand Held Terminal (HHT) interactions	<u>113140</u>
11	Downloading firmware images to Devices	<u>120147</u>
11.1	Introduction – informative	<u>120147</u>
11.2	Common Requirements	<u>121148</u>
11.3	CS05a Distribute Firmware to Communications Hub	<u>123120</u>
11.4	CS05b Distribute Firmware to ESME / GSME	<u>124124</u>
11.5	CS06 Activate Firmware	<u>126123</u>
12	Requirements for Certificates	<u>130127</u>
12.1	Requirements applicable to all Certificates	<u>130127</u>
12.2	Requirements applicable to Organisation Certificates only	<u>131128</u>
12.3	Requirements applicable to Certificates where RemotePartyRole = root or issuingAuthority	<u>131128</u>
12.4	Requirements applicable to Certificates where RemotePartyRole is neither root nor issuingAuthority	<u>132129</u>
12.5	Requirements applicable to Device Certificates	<u>132129</u>
12.6	Device processing of Certificates	<u>132129</u>
13	Managing Security Credentials on Devices	<u>134130</u>
13.1	Introduction – informative	<u>134130</u>
13.2	CS02a Provide Security Credential Details Command and Response	<u>136132</u>
13.3	CS02b Update Security Credentials Command, Response and Alert	<u>148144</u>
13.4	CS02c Issue Security Credentials	<u>184180</u>
13.5	CS02d Update Device Certificates on Device	<u>189185</u>
13.6	CS02e Provide Device Certificates from Device	<u>194190</u>
13.7	Pair-wise Authorisation of Devices	<u>199195</u>
13.8	GCS59 / 62 GPF Device Log Backup and Restore	<u>217242</u>
14	Apply Prepayment Top Up to an ESME or GSME	<u>224219</u>
14.1	Defined Terms	<u>224219</u>
14.2	Description – informative	<u>224219</u>
14.3	Common Requirements	<u>225220</u>
14.4	CS01a Applying a Prepayment Top Up to an ESME without consumer intervention <u>227222</u>	
14.5	CS01b Applying a Prepayment Top Up to a GSME without consumer intervention <u>229224</u>	
14.6	Applying a Prepayment Top Up to an ESME or GSME with consumer entry of a numeric code on the ESME or GSME	<u>230225</u>
14.7	Applying a Prepayment Top Up to an ESME or GSME with consumer entry of a numeric code on a PPMID	<u>233228</u>
14.8	Calculating and Verifying the UTRN Check Digit	<u>235230</u>
15	Message Codes	<u>237232</u>
16	Event / Alert Codes and related requirements	<u>238233</u>
16.1	Introduction – informative	<u>238233</u>
16.2	Event and Alert Codes	<u>239234</u>
16.3	Event Logs	<u>239234</u>
16.4	Requirements	<u>239234</u>

17	Remote Party Usage Rights	<u>242237</u>
17.1	Remote Party Access Rights to Attributes and Methods.....	<u>242237</u>
17.2	Remote Party Usage Rights to Use Cases	<u>242237</u>
18	Message Templates	<u>243238</u>
18.1	GBZ and ZSE Message Templates	<u>243238</u>
18.2	DLMS COSEM Message Templates	<u>246244</u>
18.3	Illustrative command and response instantiation and DER encoding.....	<u>270264</u>
18.4	Cryptographic Test Vectors	<u>281274</u>
19	Use Cases.....	<u>299289</u>
19.1	Use Case Title.....	<u>299289</u>
19.2	Use Case-specific content.....	<u>301294</u>
19.3	Embedded Use Cases.....	<u>302292</u>
20	Mapping Table.....	<u>303293</u>
21	Glossary	<u>304294</u>
22	Annex 1 – Additional DLMS Class	<u>319309</u>
22.1	Attribute description.....	<u>319309</u>
22.2	Method description	<u>320340</u>
23	Annex 2 – Counters and their use in transaction identification and Protection Against Replay – informative	<u>322344</u>
24	Annex 3 – ASN.1 modules – informative.....	<u>324343</u>
25	Annex 4 – Use of ZigBee in GBCS – informative	<u>345334</u>
25.1	Purpose.....	<u>345334</u>
25.2	GBCS requirements to use ZigBee	<u>345334</u>
25.3	GBCS requirements not to use ZigBee / vary from it	<u>345334</u>
26	Annex 5 – Use of DLMS COSEM in GBCS – informative.....	<u>347336</u>
26.1	Purpose.....	<u>347336</u>
26.2	GBCS requirements to use DLMS COSEM	<u>347336</u>
26.3	GBCS requirements not to use DLMS COSEM / vary from it.....	<u>347336</u>
27	Annex 6 – Deducing the UTRN Counter from the Truncated UTRN Counter – informative	<u>349338</u>
28	Annex 7 – Data Item Values to be set prior to installation of Devices.....	<u>351340</u>

1 Introduction – normative

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

The versions of SMETS and CHTS to which this version of GBCS is relevant are identified in the Smart Energy Code Section A.

ESME shall be certified by CESG 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 CESG 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 CESG 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 Specification (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, it shall be certified by CESG 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 Specification (PPMIDTS) requires that PPMIDs meet the requirements described in the GBCS.

This document has been brought into force by the Secretary of State on 8 November 2016. GBCS was notified to the European Commission in accordance with the requirements of the Technical Standards and Regulations Directive¹ laying down a procedure for the provision of information in the field of technical regulations and rules on Information Society services.

¹ GBCS was notified (2014/0378/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.1²) specified structures defined in this document, or in IETF RFC 5912³. Definitions of such ASN.1 structures are not repeated in the Glossary.

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

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

- the Device Specifications (SMETS (including the IHDTs, HCalCSTs and PPMIDTs), and CHTS):
 - lay out minimum physical requirements and minimum functional capabilities for Devices;
 - specify that all Devices must use the ZSE protocol specifications; and
 - specify that Electricity Smart Metering Equipment (ESME) must additionally use DLMS COSEM protocol specifications.
- International Standards documents, including those which lay out what is required to use ZSE and DLMS COSEM protocols. However, the standards are flexible and could be used in many different ways to implement technically the minimum functional requirements of SMETS and CHTS;
- the end to end protocol that is defined in the GBCS, 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 single, consistent, technical implementation of the capabilities laid out in SMETS and CHTS across all Devices, in so far as the network communications with Devices are concerned, be those communications over the Smart Metering Home Area Network (SMHAN) or Wide Area Network (WAN). The Devices in scope of this GBCS are:

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

² <http://www.itu.int/rec/T-REC-X.680-X.693-201508-I/en>

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

- 72 • Communications Hub, Communications Hub Function (CHF) and Communications
73 Hub, Gas Proxy Function (GPF);
 - 74 • Prepayment Interface Device (PPMID) and HAN Connected Auxiliary Load Control
75 Switch (HCALCS); and
 - 76 • Type 2 Devices, including In Home Displays (IHDs).
- 77 The purpose of this GBCS, and related documents, is to specify the single, consistent
78 technical implementation in sufficient detail to achieve operational interoperability of Devices.
- 79 The Smart Metering technical and security architecture is based on a suite of agreed, open
80 standards, reflecting the UK Government strategy to facilitate the development of third party
81 innovative solutions for consumer devices. These include standards relating to DLMS
82 COSEM, ZSE, ASN.1, NIST cryptography and X.509 related IETF RFCs. The GBCS does
83 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. Most Messages pass through Communications Hubs unaltered, save for any 'wrapping' information needed for transport purposes. The only exceptions are where a Communications Hub Device 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 SMETS and CHTS;
- 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 Device 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 Device Specifications.

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

There are two classifications of Message:

- HAN Only Message⁴, 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

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

- 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-XDR⁵ and Distinguished Encoding Rules (DER)⁶;
- '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

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

For example:

```
X = 0xCAFE
Y = 0xBEEF
X  $\parallel$  Y = 0xCAFE BEEF
```

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

⁵ IEC 61334-6

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

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. An overview of each of these counters and their use is included as Section 23.

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

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

4.3.1.2 Originator Counter

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

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

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.

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 Request, (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 Authority. 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`, `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*⁷' 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*⁸' 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`;
- 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:

⁷ <https://www.iad.gov/iad/library/ia-guidance/ia-solutions-for-classified/algorithm-guidance/suite-b-implementers-guide-to-fips-186-3-ecdsa.cfm>

⁸ <https://www.iad.gov/iad/library/ia-guidance/ia-solutions-for-classified/algorithm-guidance/suite-b-implementers-guide-to-fips-186-3-ecdsa.cfm>

- 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)
}
```

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	GSME	CH (CHF)	CH (GPF ⁹)	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	✓	✓		✓	✓	

⁹ 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)

4	supplier	keyAgreement	management	✓	✓		✓		
5	supplier	keyAgreement	prePaymentTopUp	✓	✓				
6	networkOperator	digitalSignature	management	✓			✓		
7	networkOperator	keyAgreement	management	✓			✓		
8	accessControlBroker	digitalSignature	management			✓			✓
9	accessControlBroker	keyAgreement	management	✓	✓	✓	✓	✓	✓
10	transitionalCoS	digitalSignature	management	✓	✓		✓	✓	
11	wanProvider	digitalSignature	management			✓			

Table 4.3.2.5: Requirements for Trust Anchor Cells by Device Type

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

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

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

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

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

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

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

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

TrustAnchorCellIdentifier			Usage of the Public Key in the Trust Anchor Cell
remotePartyRole	keyUsage	cellUsage	
root	keyCertSign	management	Used only in Certification Path Validation to check that Certification Authority Certificates and Certificates related to change of root credentials were validly issued
recovery	digitalSignature	management	Used only to verify recovery's signature on Update Security Credentials Commands addressed to the Device

TrustAnchorCellIdentifier			Usage of the Public Key in the Trust Anchor Cell
remotePartyRole	keyUsage	cellUsage	
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 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

516 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 5280¹⁰ 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 Case CS02b (Update Security Credentials).

Where a Device has successfully completed all required Command Authenticity and Integrity checks on a Command of type covered by Use Case CS02b it has received, 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 Response notifying that the Command was unsuccessful.

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*¹¹, as the Hash function;

¹⁰ <http://datatracker.ietf.org/doc/rfc5280/>

¹¹ <http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf>

- use the AES-128 cipher, as specified in *FIPS 197*¹², as the block cipher primitive;
- use the Galois Counter Mode (GCM) mode of operation as specified in *NIST Special Publication 800-38D*¹³;
- 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);
- 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 D.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*¹⁴ 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:

¹² <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>

¹³ <http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf>

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

- 644 • the parts of the Message to be signed, as defined in Section 7.2.7;
- 645 • the Private Key that the Smart Metering Entity will use in the Digital Signature
- 646 generation; and
- 647 • 0x00.

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

650 **4.3.3.3 Calculating unique Shared Secret Keys for a Remote Party Message Instance**

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

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

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

660 where:

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

664 **4.3.3.4 Calculating the Initialization Vector for GCM and GMAC**

665 In relation to Remote Party Messages, Smart Metering Entities shall use a 96 bit Initialization
 666 Vector (IV) for the GCM and GMAC algorithms as defined in *NIST Special Publication 800-*
 667 *38D*. The IV shall be the concatenation:

668 *FixedField* || *InvocationField*

669 where:

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

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

675 **4.3.3.4.1 Other input parameters to MAC and Encryption / Decryption operations –** 676 *informative*

677 Other input parameters for MAC, Encryption and Decryption are not specified in this Section
 678 4.3.3 because they vary dependent on a number of factors. These other input parameters
 679 are listed in tables of the same format as Table 4.3.3.4.1 and their values are specified in
 680 each part of the GBCS where such an operation is specified.

681 The template for such tables is the Table 4.3.3.4.1. Please note that this table does not
 682 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		

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

683 Table 4.3.3.4.1: Template for other input parameters

684 [4.3.3.4.2 Size of MAC](#)

685 The bit length of the MAC shall be 96 except for the MAC contained in the
686 WrappedApexContingencyKey extension within root Certificates, where the bit length of the
687 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.

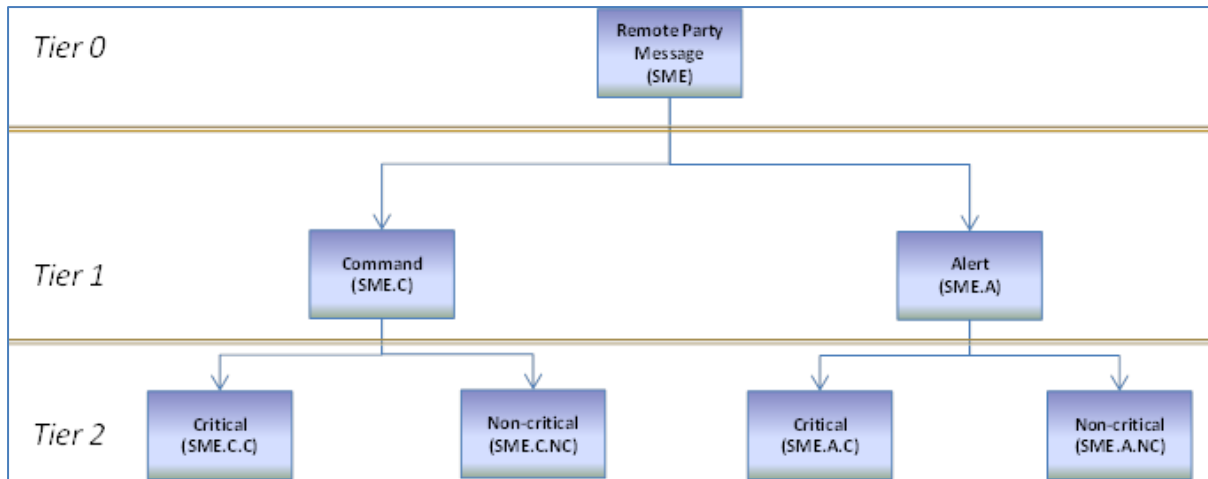


Figure 5.3: Message Categories

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 Messages¹⁵. 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

¹⁵ See Mapping Table for identification of Variant Messages

734 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

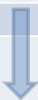
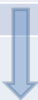
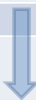




735 Table 5.4b: Common stages for Responses

736 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


737 Table 5.4c: Common stages for Alerts

738 The generic processing applied to Commands and their Responses (in relation to integrity,
739 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	<i>Not known to Device</i>	<i>Known Remote Party</i>	<i>Access Control Broker</i>	<i>Smart Metering Device</i>	<i>Smart Metering Device</i>	<i>Smart Metering Device</i>	<i>Remote Party as named in the response</i>
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

740 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	<i>Smart Metering Device</i>	<i>Smart Metering Device</i>	<i>Remote Party as named in the response</i>
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	<i>PLUS:</i> 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	<i>PLUS:</i> Can verify the MAC

Table 5.4e: Generic Alert processing

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

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

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

Note that only those parts of the sequence diagrams within yellow notes boxes are within the scope of the GBCS. The steps outside such boxes are provided for context.

For DCC managed Devices, a company called the Data Service Provider (DSP) would operate the services that provide (1) Access Control Broker, (2) Transform Service and (3) Transitional Change of Supplier. Companies called the Communication Services Providers (CSPs) would fulfil the role of WAN Provider.

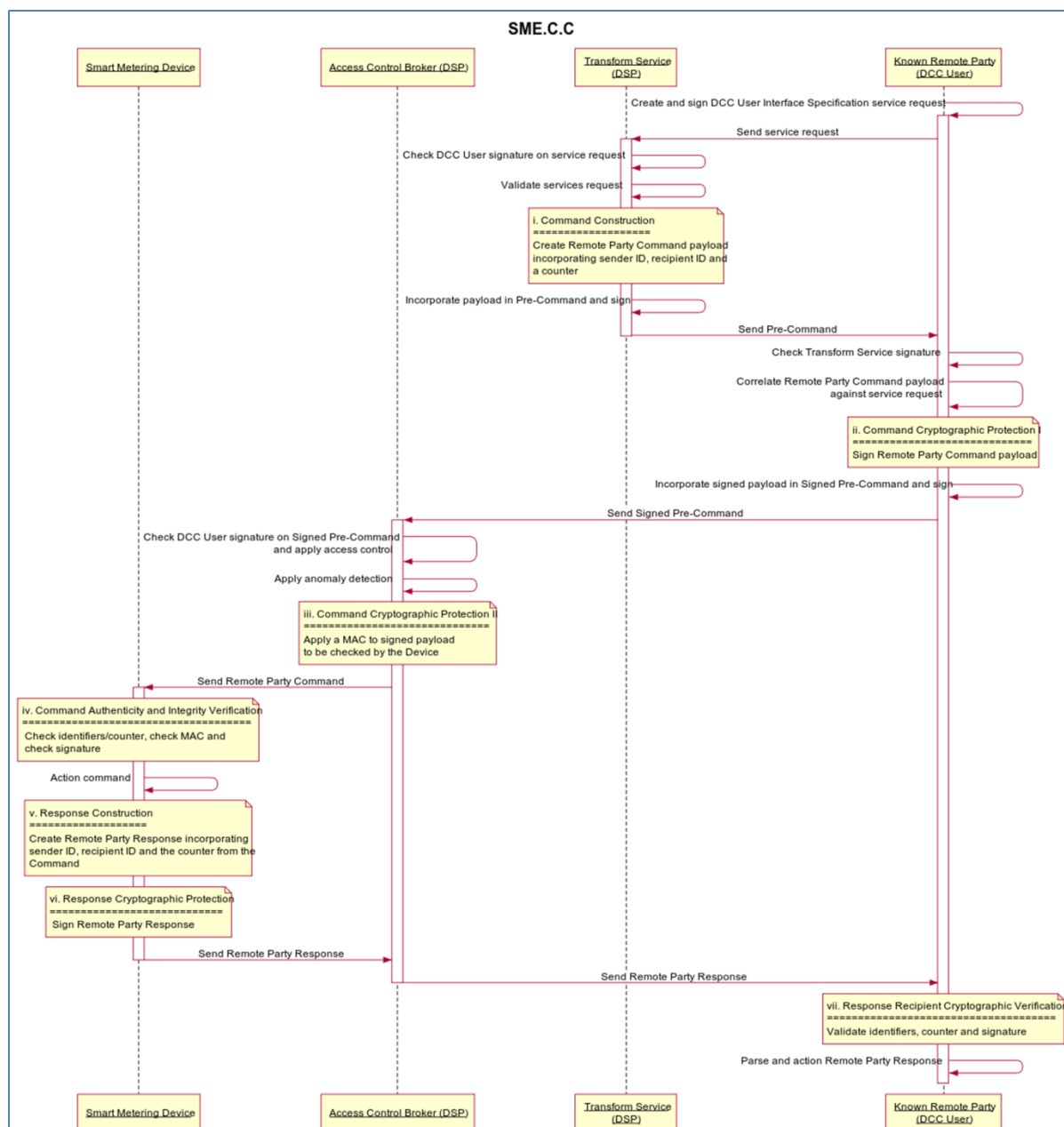


Figure 5.5a: Sequence diagram for processing Critical Remote Party Commands and Responses

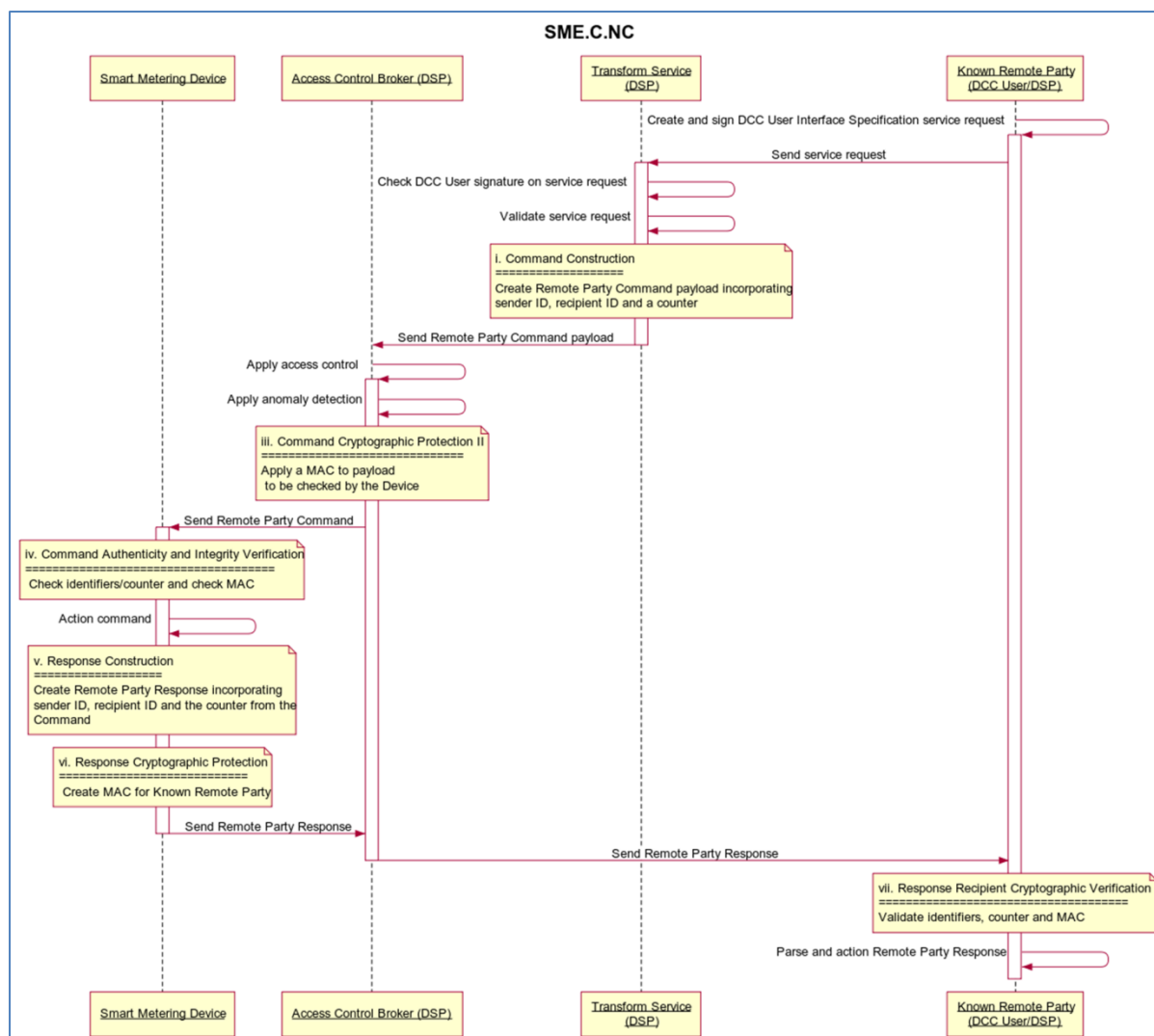


Figure 5.5b: Sequence diagram for processing non Critical Remote Party Commands and Responses

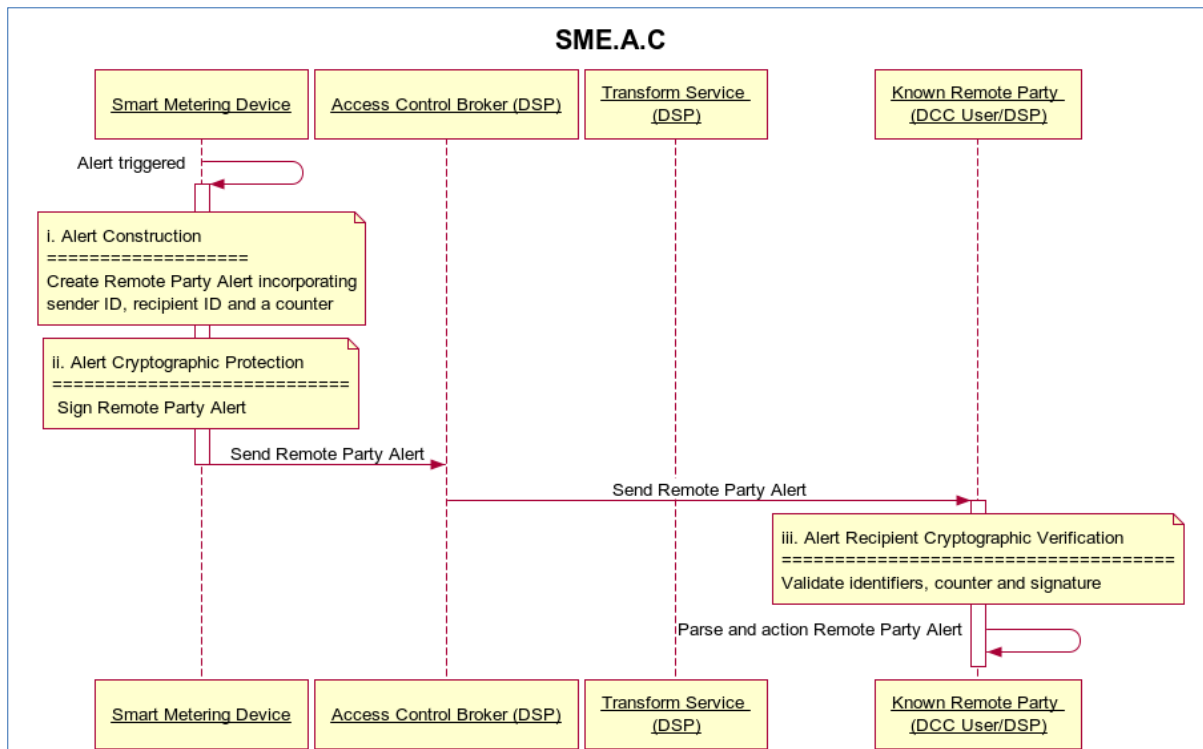


Figure 5.5c: Sequence diagram for processing Critical Remote Party Alerts

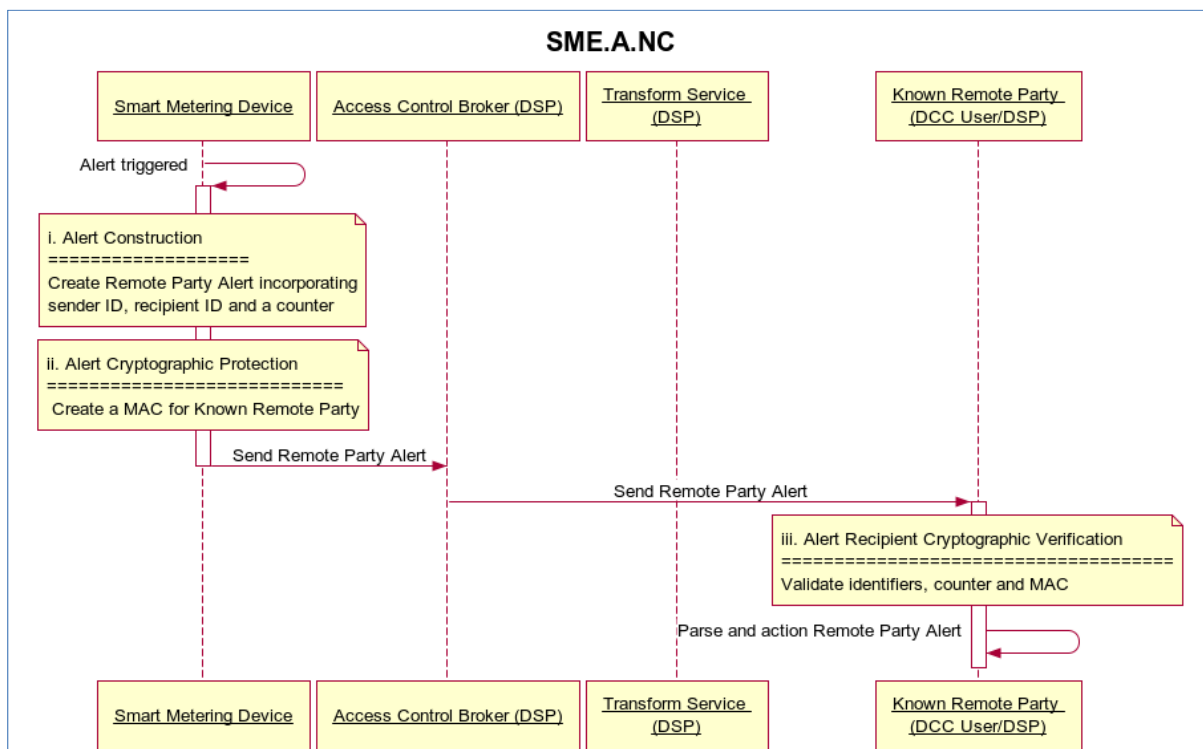


Figure 5.5d: Sequence diagram for processing non Critical Remote Party Alerts

6 Message Categories

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

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

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

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

6.1 Introduction – informative

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

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
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:	<p>Where a KRP Signature is present: 0x110000000000 Grouping Header Command Payload 0x40 KRP Signature</p> <p>Where a KRP Signature is not present: 0x110000000000 Grouping Header Command Payload 0x00</p>	

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 before undertaking any other processing of the Command.

6.2.4.1.1 Message Identifier Validation

The Device shall verify that:

- the Business Target ID in the Command has the same value as the Device's Entity Identifier;
- the Message Code is for a Message that the Device is capable of processing, according to the associated Use Case;
- 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
- 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: 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 Message requires 'Protection Against Replay' according to the corresponding Use Case, the Device shall ensure that the Originator Counter in the Command has a value that is greater than the value held by the Device for this type of Command in the corresponding Execution Counter.

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

- 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:

- 907 • a subordinate Message Category of Message Category SME.C;
 - 908 • from or to a Remote Party; and
 - 909 • a Critical Message.
- 910 A Device shall only be capable of processing the Critical Commands laid out in the GBCS.
- 911 All SME.C.C Commands and any corresponding Response shall comply both with the
- 912 requirements for SME.C Messages and with the requirements of this Section 6.3 which
- 913 covers:
- 914 • Digital Signing of the Command by the Known Remote Party;
 - 915 • verification of the Digital Signature in the Command by the Device;
 - 916 • Digital Signing of the Response by the Device; and
 - 917 • verification of the Digital Signature in the Response by the Known Remote Party.

918 **6.3.2 Processing stages**

919 *6.3.2.1 Processing stages defined in the superordinate Message Category*

920 There are no requirements additional to those of the superordinate Message Category

921 (SME.C) for the Command Cryptographic Protection II stage.

922 *6.3.2.2 Processing stages defined in subordinate categories*

923 There are no requirements for the following processing stages as they are wholly defined in

924 subordinate categories:

- 925 • Command Construction; and
- 926 • Response Construction.

927 **6.3.3 Command Cryptographic Protection I**

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

929 subordinate categories.

930 The Remote Party originating the Command shall generate a Known Remote Party

931 Signature (KRP Signature) for the Command.

932 The KRP Signature, for incorporation in the Command, shall only be generated once all

933 fields of the Command Payload and Grouping Header are populated as per the requirements

934 for the Command Construction stage, for the Message in question.

935 The KRP Signature shall be calculated across those fields of Grouping Header specified in

936 Section 7.2.7 and all fields of the Command Payload, as specified in Section 7.2.7.

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

938 **6.3.4 Command Authenticity and Integrity Verification**

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

940 subordinate categories.

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

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

944 The Device shall use the Command Payload, Grouping Header and the Public Digital

945 Signing Key of the Remote Party identified by the checks in Section 4.3.2.7.2 for Digital

946 Signature verification of the KRP Signature.

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

949 **6.3.5 Response Cryptographic Protection**

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

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

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

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

958 **6.3.6 Response Recipient Verification**

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

961 **6.4 Message Category SME.C.NC**

962 **6.4.1 Definitions**

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

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

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

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

972 **6.4.2 Processing stages**

973 ***6.4.2.1 Processing stages defined in the superordinate Message Category***

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

976 ***6.4.2.2 Processing stages defined in subordinate categories***

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

- 979 • Command Construction; and
- 980 • Response Construction.

981 **6.4.3 Command Cryptographic Protection I**

982 There are no additional requirements at the Command Cryptographic Protection I stage
 983 applicable to all Messages of Message Category SME.C.NC and any subordinate Message
 984 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.		
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 Originator 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 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.		

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 6.4.6: 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	

1037 Table 6.5.5: Calculation of GSME-PPMID MAC

1038 The GSME-PPMID MAC for incorporation in the Response shall only be calculated once all
 1039 fields of the Response, except for the GSME-PPMID MAC itself, are populated as per
 1040 requirements for the Response Construction stage, for the Message in question.

1041 6.5.6 Response Recipient Verification

1042 Requirements in this Section 6.5.6 shall apply to Message Category SME.C.PPMID-GSME
 1043 and all subordinate categories.

1044 The PPMID, as identified by the Business Originator ID in the Response, shall validate the
 1045 GSME-PPMID MAC in the Response by calculating a MAC using the parameters in Table
 1046 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	

1047 Table 6.5.6: MAC calculation for GSME-PPMID MAC validation

1048 6.6 Message Category SME.A

1049 6.6.1 Definitions

1050 The superordinate Message Category for SME.A is SME.

1051 For a Message to be of Message Category SME.A it shall be an Alert from a Device which is
 1052 addressed to a Remote Party.

1053 There are no common requirements that shall be applied to all Messages of Message
 1054 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.

1089 The SMD Signature shall be calculated across those fields of Grouping Header and all fields
1090 of the Alert Payload, both as specified in Section 7.2.7.

1091 The Device shall use its Private Digital Signing Key to generate the SMD Signature.

1092 **6.7.4 Alert Recipient Verification**

1093 Requirements in this Section 6.7.4 shall apply to Message Category SME.A.C and all
1094 subordinate categories.

1095 A Remote Party may verify the SMD Signature in the Alert by using the Alert Payload,
1096 Grouping Header and the Public Digital Signing Key for the Device, as identified in the Alert.

1097 **6.8 Message Category SME.A.NC**

1098 **6.8.1 Definitions**

1099 For a Message to be of Message Category SME.A.NC it shall be:

- 1100 • a subordinate Message Category of Message Category SME.A; and
- 1101 • not a Critical Message.

1102 All SME.A.NC Messages shall comply both with the requirements for SME.A Messages and
1103 with the requirements of this Section 6.8 which covers:

- 1104 • generation by the Device of a MAC for the Alert and validation of that MAC by the
1105 intended recipient of the Alert.

1106 **6.8.2 Processing stages**

1107 **6.8.2.1 Processing stages defined in the superordinate Message Category**

1108 There are no processing stages defined in the superordinate Message Category (SME.A).

1109 **6.8.2.2 Processing stages defined in subordinate categories**

1110 There are no requirements for the Alert Construction processing stage as they are wholly
1111 defined in subordinate categories.

1112 **6.8.3 Alert Cryptographic Protection**

1113 Requirements in this Section 6.8.3 shall apply to Message Category SME.A.NC and all
1114 subordinate categories.

1115 The Device shall calculate the Device to Known Remote Party MAC (SMD-KRP MAC) using
1116 the parameters in Table 6.8.3.

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

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 Alert Payload 0x00	

1117 Table 6.8.3: Calculation of the Device to Known Remote Party MAC

1118 The SMD-KRP MAC for incorporation in the Alert shall only be calculated once all fields of
 1119 the Alert, except for the SMD-KRP MAC itself, are populated as per requirements for the
 1120 Alert Construction stage, for the Message in question.

1121 6.8.4 Alert Recipient Verification

1122 Requirements in this Section 6.8.4 shall apply to Message Category SME.A.NC and all
 1123 subordinate categories.

1124 The Remote Party, as identified by the Business Originator ID in the Alert, may validate the
 1125 SMD-KRP MAC in the Alert by calculating a MAC using the parameters in Table 6.8.4 and
 1126 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	

1127 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).

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

It should be noted that:

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

7.2 Remote Party Message construction – general

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

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

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

1169 specifications, shall be serialised into the corresponding parts of octet strings as defined in
1170 the corresponding ZSE or ZCL specification.

1171 **7.2.1 Commands**

1172 Whether a Command requires a KRP Signature is specified in the corresponding Message
1173 Category requirements in Section 6.

1174 Where a KRP Signature is required, a Remote Party Command received by a Device shall
1175 be the concatenation:

1176 MAC Header || Grouping Header || Command Payload || 0x40 || KRP Signature || ACB-
1177 SMD MAC

1178 Where a KRP Signature is not required, a Remote Party Command received by a Device
1179 shall be the concatenation:

1180 MAC Header || Grouping Header || Command Payload || 0x00 || ACB-SMD MAC

1181 A HAN Only Command from a PPMID to a GSME shall be the concatenation:

1182 MAC Header || Grouping Header || Command Payload || 0x00 || PPMID-GSME MAC

1183 **7.2.2 Responses**

1184 Whether a Response requires an SMD Signature is specified in the corresponding Message
1185 Category requirements in Section 6.

1186 Where a SMD Signature is required, a Remote Party Response shall be the concatenation:

1187 Grouping Header || Response Payload || 0x40 || SMD Signature

1188 Where a SMD Signature is not required, a Remote Party Response shall be the
1189 concatenation:

1190 MAC Header || Grouping Header || Response Payload || 0x00 || SMD-KRP MAC

1191 A HAN Only Response from a GSME to a PPMID shall be the concatenation:

1192 MAC Header || Grouping Header || Response Payload || 0x00 || GSME-PPMID MAC

1193 **7.2.3 Alerts**

1194 Whether an Alert requires an SMD Signature is specified in the corresponding Message
1195 Category requirements in Section 6.

1196 Where a SMD Signature is required, a Remote Party Alert shall be the concatenation:

1197 Grouping Header || Alert Payload || 0x40 || SMD Signature

1198 Where a SMD Signature is not required, a Remote Party Alert shall be the concatenation:

1199 MAC Header || Grouping Header || Alert Payload || 0x00 || SMD-KRP MAC

1200 **7.2.4 Payload sequence and Break On Error**

1201 All Message Payloads - Command Payloads, Response Payloads and Alert Payloads -
1202 shall:

- 1203 • only be constructed in the sequence specified in the corresponding Use Case;
- 1204 • only be processed in the sequence specified in the corresponding Use Case; and
- 1205 • be processed by a recipient Device on a Break On Error basis.

1206 Where a Command Payload contains multiple instructions, processing of further instructions
1207 shall cease at the point any one instruction fails. In line with the DLMS COSEM Access
1208 Services requirements, a Response shall contain one result for each instruction in the

1209 Command, except where the instruction in the Command is a ZSE *GetDayProfiles* or
 1210 *GetWeekProfiles* command where one or more ZSE commands may be produced in the
 1211 Response in line with the ZSE specification. The corresponding result in the Response
 1212 Payload shall detail that instruction's success or failure. The Response Payload shall
 1213 explicitly detail a result for each of the subsequent instructions that were not attempted. The
 1214 results in the Response shall be in the same order as the instructions in the Command.

1215 The specific result codes shall be as specified in the relevant ZSE / DLMS COSEM
 1216 document or in this GBCS where standard-based error codes do not exist. Where execution
 1217 of instructions was not attempted due to the Break On Error requirement, the response shall
 1218 return:

- 1219 • for DLMS instructions, a *Data-Access-Result / Action-Result* of *other-*
 1220 *reason*;
- 1221 • for each such ZCL / ZSE instruction, a Default Response command where the ZCL
 1222 payload status field has a value of FAILURE (0x01), ZCL payload *Command ID* is set
 1223 to 0xFF and the ZCL header Frame Control field is set to 0x00 or 0x08, with the
 1224 Direction Sub-field being 0b0 or 0b1 in line with the ZigBee specifications.

1225 A ZSE command returning a status of 'NOT_FOUND' shall not be treated as a failure.

1226 7.2.5 Message construction – MAC Header

1227 The required components of the MAC Header shall be populated with the values as per
 1228 Table 7.2.5.¹⁷

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			

¹⁷ 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:

<i>transaction-id</i>	OCTET STRING,
<i>originator-system-title</i>	OCTET STRING,
<i>recipient-system-title</i>	OCTET STRING,
<i>date-time</i>	OCTET STRING,
<i>other-information</i>	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,

- 1251 ○ originator-system-title,
- 1252 ○ recipient-system-title,
- 1253 ○ date-time,
- 1254 ○ other-information.

1255 As required by Section 4.3.3.4, AK is a zero length octet string.

1256 Thus, the AAD to be used in MAC calculations that protect APDUs is the concatenation:

1257 0x110000000000 || information to be protected

1258 7.2.7 Message construction – Grouping Header

1259 The following shall be the required components of the Grouping Header and shall be
1260 populated with the values as per Table 7.2.7.

1261 Where a Signature is required in a message, it shall be calculated using only those attributes
1262 marked 'Yes' in the 'Input to the ECDSA calculation' column of Table 7.2.7, in the sequence
1263 they appear in the table.

1264 Thus, a KRP Signature or SMD Signature shall be calculated across the concatenation:

1265 CRA Flag || Originator Counter || Business Originator ID || Business Target ID || date-
1266 time (if present) || Message Code || Supplementary Remote Party ID (if present) ||
1267 Supplementary Remote Party Counter (if present) || Supplementary Originator Counter
1268 (if present) || Supplementary Remote Party Key Agreement Certificate (if present) ||
1269 (information to be protected)

1270 where (information to be protected) shall be:

- 1271 • the Command Payload in a Command;
- 1272 • the Response Payload in a Response; or
- 1273 • 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 2 or 18 or 26 or variable
Yes	value	Message Code Supplementary Remote Party ID Supplementary Remote Party Counter Supplementary Originator Counter Supplementary Remote Party Key Agreement Certificate	2 or 18 or 26 or variable	The Message Code shall always be present In an Alert, Supplementary Remote Party ID shall be present, if it is required by Section 16 In a Command or Response, the Supplementary Remote Party ID, Supplementary Remote Party Counter and Supplementary Originator Counter,

Grouping Header				
Input to the ECDSA calculation	DLMS COSEM Message Elements	Contents	Length (octets)	Note
				shall be present or not 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

1274 Table 7.2.7: Required components of Grouping Header

1275 **7.2.8 Message construction – ASN.1 Security Payloads**

1276 For Messages containing ASN.1 Security Payloads, the Payloads shall be constructed as
 1277 detailed in the Use Case for that Message Code (as defined by the Mapping Table).

1278 **7.2.9 Message construction – DLMS COSEM Payloads**

1279 For Messages containing DLMS COSEM payloads (as defined by the Message Code and
 1280 Use Cases in Section 19):

- 1281 • any Command Payload shall comply with the requirements of Table 7.2.9a and the
 1282 associated Use Case;
- 1283 • any Response Payload shall comply with the requirements of Table 7.2.9b and the
 1284 associated Use Case; and
- 1285 • any Alert Payload shall comply with the requirements of Table 7.2.9c and the
 1286 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

1287 Table 7.2.9a: Required components of Command Payload

1288 Elements marked in Table 7.2.9a as Use Case specific shall be populated according to the
 1289 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.

1290 Table 7.2.9b: Required components of Response Payload

1291 Elements marked in Table 7.2.9b as Use Case specific shall be populated according to the
 1292 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	<p>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).</p> <p>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</p>
	Data			
	Tag	0x12	1	Tag for LONG UNSIGNED
	Value	Alert Code	2	The Alert Code for this Alert, shall be as defined in Section 16
	Data			
	Tag	0x09	1	Tag for octet-string
	Length	0x0C	1	Twelve characters long as DLMS date times are octet-string(12)
	Value	Time Stamp	12	The time stamp for this Alert, shall be as defined in Section 16
2	Use Case Specific Additional Content	Use Case specific	Use Case	See Note at row 1, which means that, for most Alerts, there will be no Use Case specific content.

1293 Table 7.2.9c: Required components of Alert Payload

1294 Elements marked in Table 7.2.9c as Use Case specific shall be populated according to the
 1295 Use Case for the Message Code (see Section 19).

1296 7.2.10 Message construction – GBZ Payloads

1297 A GBZ Payload shall be a Payload containing one or more ZSE / ZCL commands. For
 1298 clarity, this includes Payloads in HAN Only Commands between a PPMID and a GSME.

1299 For Messages containing GBZ Payloads (as defined by the Mapping Table):

- 1300 • any Command Payload shall comply with the requirements of Table 7.2.10a and the
1301 associated Use Case;
- 1302 • any Response Payload shall comply with the requirements of Table 7.2.10b and the
1303 associated Use Case; and
- 1304 • any Alert Payload shall comply with the requirements of Table 7.2.10c and the
1305 associated Use Case.
- 1306 Each GBZ Use Case Specific Component shall comply with:
- 1307 • Table 7.2.10d if the ZSE / ZCL command within it is not encrypted; or
- 1308 • 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

1309 Table 7.2.10a: Required components of GBZ Command Payload

1310 Elements marked in Table 7.2.10a as Use Case specific shall be populated according to the
1311 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

1312 Table 7.2.10b: Required components of GBZ Response Payload

1313 Elements marked in Table 7.2.10b as Use Case specific shall be populated according to the
1314 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 <i>UTCTime</i> , with its ZCL meaning, at which this Alert was created
2	<i>GBZ Use Case Specific Component(s)</i>	<i>Use Case specific</i>		<i>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</i>

1315 Table 7.2.10c: Required components of GBZ Alert Payload

1316 Elements marked in Table 7.2.10c as Use Case specific shall be populated according to the
 1317 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 corresponding response-style ZSE / ZCL

GBZ Use Case Specific Component without encrypted content				
No	Message Elements	Contents	Length (octets)	Note
				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

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 Messages¹⁸

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:

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

1363 Message Routing Header || GBT Header || GBT Block Data

1364 where:

- 1365 • Message Routing Header shall be structured and populated according to Table
1366 7.2.11.5. Note that Message Routing Header (1) uniquely identifies the GBT
1367 Message Series, (2) identifies whether the GBT Message is being sent to or from the
1368 Device and (3) unambiguously ties all GBT Messages in the GBT Message Series to
1369 the single Remote Party Message being exchanged;
- 1370 • GBT Header shall be structured and populated according to Table 7.2.11.6; and
- 1371 • GBT Block Data shall be the part of the Remote Party Message, constructed as per
1372 Section 7.2.11.4, being carried in this GBT Message.

1373 *7.2.11.4 GBT Message processing*

1374 The GBT Initiator shall, once the Remote Party Message is fully constructed and all
1375 cryptographic protections are applied, slice the octet string produced so that:

- 1376 1. GBT Block Data with GBT Initiator Block Number of 1 is the 1149 most significant octets
1377 of the Remote Party Message, or all of the octets if the size of the Remote Party
1378 Message is less than 1149 octets;
- 1379 2. GBT Block Data with GBT Initiator Block Number of 2 is the next 1149 most significant
1380 octets of the Remote Party Message, or all of the octets if the size of the remaining
1381 octets in Remote Party Message is less than 1149 octets; and
- 1382 3. remaining GBT Block Data are created by repeating Step 2, each time incrementing
1383 GBT Initiator Block Number by 1, until there are no remaining octets in the Remote
1384 Party Message.

1385 The GBT Recipient shall not undertake any processing, in the sense of Section 6, of the
1386 Remote Party Message carried in a GBT Message Series until it has received:

- 1387 • a GBT Message in this GBT Message Series where the 'last-block' field contains 0b1
1388 (meaning last block); and
- 1389 • all GBT Messages in this GBT Message Series with 'block-number' fields less than
1390 the 'block-number' field in the last block. Where the GBT Recipient has not received
1391 all such GBT Messages, it shall send a GBT Message (Request Block Resend) for
1392 each missing block-number. Where the GBT Recipient is a Device, it may discard all
1393 blocks in a GBT Message Series if it has received no response to a GBT Message
1394 (Request Block Resend) after 60 minutes.

1395 When a GBT Message Series carries a Response and that GBT Message Series consists of
1396 more than 63 GBT Messages, the GBT Recipient shall, if it wishes to receive all parts of the
1397 GBT Message Series, send a GBT Message (Acknowledgement) when it has received each
1398 complete set of 63 sequential GBT Messages. Note, when the Device sending the
1399 Response receives such a GBT Message (Acknowledgement), it is able to begin sending the
1400 next sequence of GBT Messages in the GBT Message Series.

1401 GBT Recipient Block Number shall be set to 0x0001 in the first GBT Message sent by the
1402 GBT Recipient. It shall be incremented by 1 in each subsequent GBT Message it sends.

1403 GBT Initiator Block Number Ack shall be the highest of:

- 1404 • 0x0000; and
- 1405 • the highest block-number in any GBT Message the GBT Initiator has received in this
1406 GBT Message Series.

1407 GBT Recipient Block Number Ack shall:

- 1408 • in a GBT Message (Acknowledgement), be the highest block-number in any GBT
1409 Message the GBT Recipient has received in this GBT Message Series; and
- 1410 • in a GBT Message (Request Block Resend), the value of block-number up to which
1411 the GBT Recipient has received all the prior numbered GBT Messages in this GBT
1412 Message Series.

1413 Where the GBT Initiator is a Device, the Device shall be able to resend any GBT Message
1414 within a GBT Message Series, for a minimum period from when it sends the first GBT
1415 Message in that series, to whichever is the sooner of:

- 1416 • it receiving an authenticated GBT Message (Acknowledgement) where the GBT
1417 Recipient Block Number Ack contains a value equal to the highest value of GBT
1418 Initiator Block Number Ack in this GBT Message Series; or
- 1419 • 24 hours later.

1420 For clarity, Devices shall discard malformed GBT messages without further processing in the
1421 sense of Section 6. For example, the following GBT Messages would be malformed:

- 1422 • one with 'block-number' equal to zero (this does not comply with the GBCS); and
- 1423 • one delivered to a recipient where 'block-number-ack' is greater than the greatest
1424 'block-number' in any GBT Message the recipient has sent in that GBT Message
1425 Series.

1426 Where the GBT Recipient is an ESME, the ESME shall not send a GBT Message (Request
1427 Block Resend), in relation to a GBT Message Series, until and unless 30 seconds have
1428 elapsed since receipt of the most recent GBT Message in that GBT Message Series. If 30
1429 minutes elapse without the ESME receiving corresponding GBT Message(s) in response,
1430 the ESME may discard the GBT Messages it has received in this GBT Message Series.

1431 **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 Entity Identifier of the GBT Third Party

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

1432 Table 7.2.11.5: Message Routing Header

1433 **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.

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

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:

1453 Message Routing Header || GBT Header || MAC Header || Grouping Header || read
 1454 MPAN Response Payload || 0x00 || SMD-KRP MAC

1455 Example 2: A large Command with small Response – e.g. set tariff on an ESME where the
 1456 tariff is complex.

1457 Without GBT, the Command is:

1458 MAC Header || Grouping Header || set Tariff Command Payload || 0x40 || KRP
 1459 Signature || ACB-SMD MAC

1460 For the purposes of example, assume this is divided into three blocks, so block-numbers 1, 2
 1461 and 3. Actual set tariff Commands will vary from this number of blocks.

1462 The Command is transmitted as the GBT Message Series:

1463 Message Routing Header || GBT Header || block 1
 1464 Message Routing Header || GBT Header || block 2
 1465 Message Routing Header || GBT Header || block 3

1466 This is reconstructed at the ESME, by concatenating blocks 1, 2 and 3 to give:

1467 MAC Header || Grouping Header || set Tariff Command Payload || 0x40 || KRP
 1468 Signature || ACB-SMD MAC

1469 The Response would have the following structure:

1470 Message Routing Header || GBT Header || Grouping Header || Response Payload ||
 1471 0x40 || SMD Signature

1472 It will be smaller than the Maximum PDU Size and so can be sent as a single APDU without
 1473 any use of GBT.

1474 Example 3: small Command with large Response – e.g. read half-hourly profile (Export)

1475 The Command is:

1476 MAC Header || Grouping Header || read half-hourly profile (Export) Command Payload ||
 1477 0x00 || ACB-SMD MAC

1478 It will be smaller than the Maximum PDU Size and so can be sent as a single APDU without
 1479 any use of GBT.

1480 The ESME Response is:

1481 MAC Header || Grouping Header || read half-hourly profile (Export) Response Payload ||
 1482 0x00 || SMD-KRP MAC

1483 Assuming that there are 75 blocks to send the GBT Message Series would, if no retries are
 1484 needed, be as follows:

1485 The ESME will send:

1486 Message Routing Header || GBT Header || Block 1
 1487 ...
 1488 Message Routing Header || GBT Header || Block 63

1489 and wait for acknowledgement.

1490 The Access Control Broker will construct and send that acknowledgement whose structure
 1491 is:

1492 Message Routing Header || GBT Header

1493 When this acknowledgement is received by the ESME, the ESME will send:

1494 Message Routing Header || GBT Header || Block 64
 1495 ...

1496 Message Routing Header || GBT Header || Block 75

1497 When the whole Message is received by the Access Control Broker, the Response can then
1498 be reconstructed:

1499 MAC Header || Grouping Header || read half-hourly profile (Export) Response Payload ||
1500 0x00 || SMD-KRP MAC

1501 Once the response has been reconstructed, the MAC can be checked.

1502 7.3 Device Requirements – DLMS COSEM

1503 7.3.1 Introduction – informative

1504 The DLMS COSEM server in the ESME (and CHF where a DLMS COSEM server is
1505 present) responds to requests for information, and also provides Alerts in response to events
1506 within the meter (e.g. push data at the end of billing period; Alert in the event of a tamper;
1507 disable supply when prepayment credit expires). To achieve this, a level of configuration is
1508 needed to ensure that the behaviour of the Device is as expected.

1509 SMETS and CHTS require that the Critical Commands mandated by them (and so those
1510 defined in the GBCS) are the only Critical commands allowed. Devices may implement
1511 additional non Critical features only.

1512 SMETS and CHTS only require DLMS COSEM certification on the ESME.

1513 DLMS COSEM objects (or functionality equivalent to them) are required to deliver the ESME
1514 functionality defined in the Use Cases in a consistent way but should not be accessible via
1515 the ESME's HAN interface (i.e. it is internal functionality).

1516 7.3.2 General Requirements

1517 Constant values specified in Table 7.3.8a shall be fixed before operation and shall be
1518 immutable save via a firmware upgrade. This is to ensure consistent functioning and guard
1519 against potential attacks. Except where explicitly required by this Section 7.3, a Device shall
1520 not expose any part of any DLMS COSEM object, either for the writing of an attribute or for
1521 the invocation of a method that could, if used, constitute a Critical action.

1522 For Devices which are not ESME or CHF (so where `deviceType` <> 1 or 2), the GBCS
1523 does not require the implementation of any DLMS COSEM objects.

1524 All Devices which are ESME (so where `deviceType` = 1):

- 1525 • shall implement all of the DLMS COSEM objects, attributes and methods detailed in
1526 'SMETS Required Objects' tab of the table in Section 20, and expose the specified
1527 attributes and methods over its network interface; and
- 1528 • shall have the constant values set for the DLMS COSEM attributes specified as
1529 requiring constant values in Table 7.3.8a, and shall ensure that such values cannot
1530 be amended, save via activation of new firmware.

1531 7.3.3 Application Associations

1532 Any ESME or CHF shall communicate using pre-established Application Associations (AA).
1533 These shall be set at manufacture, and the Device shall reject all subsequent attempts to
1534 open or release Application Associations.

1535 An ESME or CHF shall support the Application Associations in Table 7.3.8c. An ESME or
1536 CHF shall not support any additional Application Associations.

1537 The Application Associations in Table 7.3.8c shall limit access to DLMS features by
1538 configuring the *object_list* attribute to reflect the access granted to the role in 'SMETS
1539 Required Objects' tab of the Mapping Table in Section 20. Any other methods and attributes

1540 of any class shall be made inaccessible by listing them in the *object_list* attribute such that
 1541 there is no access.

1542 The Public AA shall only expose:

- 1543 • the SAP Assignment object; and
- 1544 • the DLMS COSEM Logical Device name object and *object_list* with no objects listed
 1545 other than the Association Logical Name (LN) Object (with its Blue Book meaning)
 1546 and the SAP Assignment Object.

1547 When a Message is received by the ESME or CHF, the Message shall be validated against
 1548 the AA based on the Business Originator ID and the Message Code within the Grouping
 1549 Header.

1550 Other attributes in the Association LN Objects and the Security Setup Objects shall be set at
 1551 manufacture in accordance with Tables 7.3.8d and 7.3.8e.

1552 The 'SAP Assignment' object shall be configured at manufacture in accordance with Table
 1553 7.3.8f. The method associated with the 'SAP Assignment' object shall not be accessible to
 1554 any Application Association.

1555 **7.3.4 Interface Classes and Objects**

1556 Devices shall support the version of Interface Classes shown as current in the Blue Book.

1557 An ESME shall support the 'Class 9000' as detailed in Section 22 of this GBCS.

1558 Unless explicitly required in a predetermined script or the SMETS 'Required Objects' tab of
 1559 the Mapping Table, Class 3 objects shall not have a reset method that is accessible external
 1560 to the Device.

1561 Unless otherwise stated, Generic Profile objects with a non-zero attribute 4 shall capture the
 1562 first entry at midnight UTC.

1563 The ESME shall have the constant values set for the DLMS COSEM attributes specified as
 1564 requiring constant values in Table 7.3.8a, and shall ensure that such values cannot be
 1565 amended, save via activation of new firmware.

1566 **7.3.5 Values normally negotiated when an AA is established**

1567 **7.3.5.1 Conformance Block Contents**

1568 The conformance block shall be set according to Table 7.3.8g.

1569 **7.3.5.2 Other Items.**

1570 Other items for pre-establishing the Application Associations and other communication
 1571 parameters shall be implemented as detailed in Table 7.3.8h.

1572 **7.3.5.3 Security Setup Objects**

1573 Security Setup Objects shall be limited to those listed in Table 7.3.8c.

1574 Manufacturer specific attributes and methods for these objects shall not be accessible
 1575 external to the Device.

1576 The methods of the Security Setup objects shall not be accessible external to the Device.
 1577 The attributes of the Security Setup objects shall be as specified in Table 7.3.8e.

1578 Note that Security Credentials are updated as specified in Section 13.

1579 **7.3.6 Scripts for operation of the meter**

1580 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.

7.3.7 Pricing matrices, scripts and registers

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	

Description	Script Selector (long unsigned)	Consumption Register	Charge Object	Index (octet-string(1))	Notes
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	
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	

Description	Script Selector (long unsigned)	Consumption Register	Charge Object	Index (octet-string(1))	Notes
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

Description	Script Selector (long unsigned)	Consumption Register	Charge Object	Index (octet-string(1))	Notes
					ESME consumption since last block reset
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
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

Description	Script Selector (long unsigned)	Consumption Register	Charge Object	Index (octet-string(1))	Notes
					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
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

Description	Script Selector (long unsigned)	Consumption Register	Charge Object	Index (octet-string(1))	Notes
					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

1618 Table 7.3.7.2: ESME requirements for pricing matrices, scripts and registers

1619 7.3.8 DLMS Device Requirements Tables

1620 Table 7.3.8a: Objects tab in embedded file

1621 Table 7.3.8b: Scripts tab in embedded file

1622 Table 7.3.8c: Application Associations tab in embedded file

1623 Table 7.3.8d: Association LN Object Content tab in embedded file

1624 Table 7.3.8e: Security Setup Object Content tab in embedded file

1625 Table 7.3.8f: SAP Assignment Object content tab in embedded file

1626 Table 7.3.8g: Conformance Content tab in embedded file

1627 Table 7.3.8h: End to End Communications tab in embedded file



GBCS v1.0 DLMS
Device Requirements

1628

7.3.9 ESME accounts, credits and charges – informative

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 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 shall use the corresponding Specials Days Object in Table 7.3.10.

SMETS Reference	Calendar / Scheduler object		Special Days Object to be used
	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
AuxiliaryLoadControlSwitchesCalendar(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.

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)'.

- 1665 A GPF shall implement a *ZSE Energy Services Interface* and shall implement all the
 1666 *clusters, commands, attribute sets and attributes* in Table 7.4 where column A is 'GPF:
 1667 Energy Services Interface (Gas ESI Endpoint)'
- 1668 A CHF shall implement a *ZSE Remote Communications Device* and shall implement all the
 1669 *clusters, commands, attribute sets and attributes* in Table 7.4 where column A is 'CHF:
 1670 Remote Communications Device (Remote Communications Endpoint)'
- 1671 An ESME which is not a Twin Element ESME shall implement a *ZSE Energy Services*
 1672 *Interface* and shall implement all the *clusters, commands, attribute sets and attributes* in
 1673 Table 7.4 where column A is 'ESME: Energy Services Interface (Electricity ESI Endpoint)'
- 1674 An ESME which is a Twin Element ESME shall implement three *ZSE Energy Services*
 1675 *Interfaces*:
- 1676 1. the first which shall implement all the *clusters, commands, attribute sets and attributes*
 1677 in Table 7.4 where column A is 'ESME: Energy Services Interface (Twin ESME
 1678 aggregate ESI Endpoint)';
 - 1679 2. the second which, in relation to the primary measuring element, shall implement all the
 1680 *clusters, commands, attribute sets and attributes* in Table 7.4 where column A is 'ESME:
 1681 Energy Services Interface (Twin ESME primary/secondary ESI Endpoint)'; and
 - 1682 3. the third which, in relation to the secondary measuring element, shall implement all the
 1683 *clusters, commands, attribute sets and attributes* in Table 7.4 where column A is 'ESME:
 1684 Energy Services Interface (Twin ESME primary/secondary ESI Endpoint)'
- 1685 A PPMID shall implement a *ZSE In-Home Display* and shall implement all the *clusters,*
 1686 *commands, attribute sets and attributes* in Table 7.4 where column A is 'PPMID: In-Home
 1687 Display'.
- 1688 An HCALCS shall implement a *ZSE Load Control Device* and shall implement all the
 1689 *clusters, commands, attribute sets and attributes* in Table 7.4 where column A is 'HCALCS:
 1690 Load Control Device'.
- 1691 An HHT shall implement a *ZSE Remote Communications Device* and shall implement all the
 1692 *clusters, commands, attribute sets and attributes* in Table 7.4 where column A is 'HHT:
 1693 Remote Communications Device'.
- 1694 An IHD shall support the mandatory attributes of the Basic cluster and the other clusters,
 1695 attributes and commands necessary to meet the SMETS requirements.
- 1696 Where a row in Table 7.4 is required for a Device, that Device shall support the cluster,
 1697 attribute or command specified in that row as client or server, as specified in column C
 1698 (labelled 'Client / Server').
- 1699 Support for *clusters, commands, attribute sets and attributes* shall be as defined in columns
 1700 B ('Cluster'), D ('Command'), E ('Attribute Set') and F ('Attribute').
- 1701 Note that the other columns in Table 7.4 are informative and for requirements traceability
 1702 only.
- 1703 Except where explicitly required by this Section 7.4 or by Section 19.3, a Device shall not
 1704 execute any ZSE command, be that in a GBZ Command Payload or provided as a native
 1705 ZSE command, that could, if executed, constitute a Critical action. For clarity, a Device shall
 1706 not execute a ZSE *Publish Change of Supplier* command if bits 11-12 of the *Provider*
 1707 *Change Control* parameter (*Meter Contactor State*) of that command has any value other
 1708 than 0b11 (*Supply UNCHANGED*).



GBCS v1.0 Device
Requirements.xlsx

1709

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.

1750 Where all the octets in the 'from' date-time are 0x00 (excluding the least significant 3 bytes
 1751 in Blue Book octet string formatted date-times), the Device shall interpret the 'from' field as
 1752 meaning from the oldest in the Log.

1753 Where all the octets in the 'to' date-time are 0xFF (excluding the least significant 3 bytes in
 1754 Blue Book octet string formatted date-times), the Device shall interpret the 'to' field as
 1755 meaning to the newest in the Log.

1756 Where the Encrypted data item in the Mapping Table is not in a Log, a Command requesting
 1757 that data shall never have 'from' or 'to' date-times specified.

1758 **8.3 Key Derivation Inputs**

1759 Where a Remote Party Message (1) contains Encrypted data items and (2) contains a
 1760 Supplementary Remote Party ID, then the Encryption Remote Party shall be that identified
 1761 by the Supplementary Remote Party ID. Otherwise, the Encryption Remote Party shall be
 1762 the Remote Party identified in the Grouping Header of the Message.

1763 If the Message is to include a Supplementary Originator Counter generated by the Device
 1764 (see Section 4.3.1.4), then the Encryption Originator Counter shall be the Supplementary
 1765 Originator Counter. Otherwise the Encryption Originator Counter shall be the Originator
 1766 Counter with the value in the Grouping Header of the Message.

1767 In relation to the Key Derivation Function requirements at Section 4.3.3, fields shall be
 1768 populated as follows:

- 1769 • 'value of transaction-id' shall be the concatenation 0x04 || Encryption Originator
 1770 Counter. Note 0x04 ensures this value is not used in any other Key Derivation
 1771 Function invocation save that related to this Encryption / Decryption; and
- 1772 • for Encrypted data items in Responses and Alerts, 'value of recipient-system-title'
 1773 shall be Encryption Remote Party and 'value of originator-system-title' shall be the
 1774 Device's Entity Identifier.

1775 **8.4 AAD, Plaintext and Ciphertext**

1776 The Plaintext shall be set to the structure and content of the data item(s) as they would have
 1777 been exposed on the Device's HAN interface, if access to them were not constrained to be
 1778 via Encrypted form by this Section 8.4.

1779 AAD shall be set to security control byte (SC) which shall have the value of 0x31 (see
 1780 Section 8.4.1).

1781 The Invocation Counter (IC) shall have a value of 0x00000000.

1782 The Authenticated Encryption MAC (AE MAC) shall be the MAC produced by applying
 1783 Authenticated Encryption to AAD and Plaintext, as defined in *NIST Special Publication 800-38D*,
 1784 with the values specified in this Section 8.4.

1785 Authenticated Encryption (AE) Ciphertext shall be the Ciphertext produced by applying
 1786 Authenticated Encryption to Plaintext, with the values specified in this Section 8.4.

1787 Ciphred Information shall be the concatenation:

1788 SC || IC || AE Ciphertext || AE MAC

1789 **8.4.1 Meaning of SC – informative**

1790 The SC is set to 0x31 to reflect the following:

- 1791 • Bits 3..0 are security suite which is 0b0001 since Security Suite 1 is required;
- 1792 • Bit 4 is set to 0b1 since Authentication of the data is required;

- 1793 • Bit 5 is set to 0b1 since Encryption of the data is required;
- 1794 • Bit 6 is set to 0b0 since Messages containing the encrypted data are unicast; and
- 1795 • Bit 7 is set to 0b0 as per the Green Book.

1796 8.5 Access to sensitive data – COSEM attribute access

1797 Access to sensitive data items shall be via the Data Protection class, as specified in the Blue
 1798 Book. The required OBIS codes and associated details for each attribute shall be as
 1799 specified in the 'SMETS required objects' tab in the Mapping Table.

1800 The Device shall only allow read access to attributes listed as Encrypted in the Mapping
 1801 Table using the get_protected_attributes(data) method of the Data Protection class and not
 1802 allow access to any other methods of such objects.

1803 8.5.1 Values of the Data Protection class attributes

1804 The values of attributes 1, 3, 4, 5 and 6 of an object of the Data Protection class shall be set
 1805 on a Device at manufacture, and those values shall not be capable of amendment except by
 1806 firmware upgrade. For each object of the Data Protection class:

- 1807 • protection_object_list (attribute 3) shall be an array containing object_definition(s).
 1808 Within an object_definition: class_id, logical_name, attribute_index, data_index,
 1809 restriction_type and restriction_value shall take values as per Table 7.3.8a;
- 1810 • the value of protection_parameters_get (attribute 4) shall be a single entry array
 1811 containing one protection_parameters_element, which shall have the values
 1812 specified in Table 8.5.1;
- 1813 • the value of protection_parameters_set (attribute 5) shall be an array containing zero
 1814 entries; and
- 1815 • the value of required_protection (attribute 6) shall be 0b01100000 (0x60) where the
 1816 object exposes the get_protected_attributes(data) method, since Authenticated
 1817 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_ciphred_data	octet-string	An octet string of length zero

1819 Table 8.5.1: Values of protection_parameters_element

1820 8.5.2 Parameters of the get_protected_attributes method

1821 The get_protected_attributes_request parameter of the get_protected_attributes method
 1822 shall:

- 1823 • be populated in the Command to the Device according to Table 8.5.2a; and
- 1824 • be verified by the Device receiving the Command according to Table 8.5.2a.
- 1825 The protection_parameters part of the get_protected_attributes_response returned by the
- 1826 get_protected_attributes method shall be populated by the Device according to Table 8.5.2b.
- 1827 The value of protected_attributes part of the protected_attributes_response_data returned by
- 1828 the get_protected_attributes method shall be populated by the Device with Ciphared
- 1829 Information, calculated as per the requirements of Section 8.2. The tag for
- 1830 protected_attributes shall be 'octet-string' (0x09) and the length shall be the length of
- 1831 Ciphared Information.

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

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

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

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

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

Field	Value	Device Validation	Note
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	Must be the Device's Entity Identifier	The Entity Identifier of the Encryption Remote Party
other_information			
tag	0x09	Must have this value	Meaning 'octet-string'
length	0x00	Must have this value	Zero length since this string is empty
key_info		Structure	
tag	0x02	Must have this value	Meaning 'structure'
length	0x02	Must have this value	2 elements in the structure
key_info_type:			
tag	0x16	Must have this value	Meaning 'enum'
value	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_ciphred_data			
tag	0x09	Must have this value	Meaning 'octet-string'
length	0x00	Must have this value	Zero length since this string is empty

1834 Table 8.5.2b: values of protection_parameters

1835 8.5.3 Billing Data Log Alert – DLMS COSEM

1836 'Use Case Specific Additional Content' within the Billing Data Log Alert shall be populated
 1837 according to the Message Template for the Billing Data Log Alert in Section 18.2.

1838 **8.6 Access to sensitive data – ZSE attribute access**

1839 Ciphared Information shall be used to populate each 'GBZ Use Case Specific Component
1840 with encrypted content' as specified in Section 7.2.10.

9 Time Synchronisation and Future Dated Remote Party Messages

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

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

9.1 Time synchronisation

9.1.1 Introduction – informative

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

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

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

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

This leads to a time synchronisation approach for ESME as specified in Section 9.1.4, and GSME as specified in Section 9.1.7.

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

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; 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';

- 1921 • if the time provided by the Communications Hub differs from the Device's time by
 1922 more than 10 seconds, then the Device shall:
- 1923 ○ set its Time Status to 'Unreliable'; and
- 1924 ○ if this results in a change to Time Status, the Device shall construct and issue an
 1925 Alert with Alert Code 0x8F0C, meaning that its time would have been shifted by
 1926 more than 10 seconds. If the Device is a GSME, the Alert Payload shall be a GBZ
 1927 Alert Payload. If the Device is an ESME, the Alert Payload shall be a DLMS
 1928 COSEM Alert Payload.
- 1929 • if the time provided by the Communications Hub differs from the Device's time by 10
 1930 seconds or less, then the Device shall adjust its time to the Communications Hub's
 1931 time.

1932 9.1.4 ECS70 Set Clock on ESME

1933 This Use Case covers the setting of the Clock by the Supplier on an ESME.

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

1934 Table 9.1.4: Use Case Cross References for ECS70 Set Clock on ESME

1935 9.1.4.1 Pre-conditions

1936 None.

1937 9.1.4.2 Detailed Steps

1938 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 0xFFFFFFFFFFFFFFFF8000FF 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

1939 Table 9.1.4.2a: Construction of Command Payload

1940 In this Section 9.1.4.2, the object with OBIS code 0-0:1.0.0.255 shall be referred to as the
 1941 *Clock* and italicised terms shall have their Blue / Green Book meaning.

1942 On receipt of the Command, the ESME shall undertake processing in the following
 1943 sequence:

- 1944 1. the ESME shall undertake the 'Command Authenticity and Integrity Verification'
 1945 processing required for Commands of type SME.C.C. If that fails, processing shall
 1946 cease;
- 1947 2. the ESME shall process the instructions in the *access-request-body* of the Command as
 1948 follows:
- 1949 a) when attribute 9 of the *Clock* (*clock_base*) is set to '*radio controlled*' (5), the
 1950 ESME shall request the value of the Communications Hub Time from the
 1951 Communications Hub via its ZigBee radio. If a time of 0xFFFFFFFF is
 1952 provided or if no time is received:
- 1953 i. if the current Time Status is set to 'Reliable', the ESME shall set bit 1 of
 1954 attribute 4 (so setting *status* of Clock to be '*doubtful value*' (Unreliable)); or

- 1955 ii. if the current Time Status is not set to 'Reliable', the ESME shall not
1956 change status.
- 1957 b) the *preset_adjusting_time* method of the Clock shall be executed. Note this is
1958 to set parameters for the *adjust_to_present_time* method;
- 1959 c) the *adjust_to_present_time* method of the Clock shall be executed as follows:
- 1960 iii. if the Communications Hub Time returned lies between
1961 *validity_interval_start* and *validity_interval_end*, then:
- 1962 a. ESME *time* shall be updated to match Communications Hub Time;
- 1963 b. the ESME shall unset bit 0 of attribute 4 (so setting *status* of the *Clock*
1964 not to be an 'invalid value'); and
- 1965 c. the ESME shall unset bit 1 of attribute 4 (so setting *status* of the *Clock*
1966 not to be a 'doubtful value').
- 1967 iv. if the Communications Hub Time returned lies before *validity_interval_start*
1968 (Supplier Current Time) or after *validity_interval_end* (Supplier Current
1969 Time + Tolerance)) and (bit 0 of attribute 4 of the *Clock* is unset), then:
- 1970 d. time shall remain unchanged, since time is outside the
1971 *validity_interval*; and
- 1972 e. the ESME shall set bit 1 of attribute 4 and unset bit 0 of attribute 4 (so
1973 setting *status* of *Clock* to be a 'doubtful value' (Unreliable)).
- 1974 d) the *get request* on the *time* and *status* attributes of the *Clock* shall be executed.
- 1975 3. the ESME shall undertake the 'Response Construction' and 'Response Cryptographic'
1976 processing required for a Response of type SME.C.C.
- 1977 On receipt of the Response, the recipient may undertake the 'Response Recipient
1978 Processing' for Responses of type SME.C.C.
- 1979 The meaning of result attributes is as defined in the Green Book.
- 1980 The meaning of the unsigned integer returned by the *get request* on attribute 4 of the Clock
1981 (*status*) is as per Table 9.1.4.2b.

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

1982 Table 9.1.4.2b: Meaning of unsigned integer

1983 9.1.5 Time related object on ESME

1984 Italicised terms in this section shall have their Blue Book meaning.

1985 An ESME shall have a *Data* object with *OBIS* code 0-0:94.44.100.255 where attribute 2 of
1986 that object:

- 1987 • shall be a double-long-unsigned value;
- 1988 • shall have a value set by the ESME to the number of seconds between 0 hours 0
1989 minutes 0 seconds on 1st January 2000 UTC and the value of UTC time specified by
1990 attribute 2 of the *Clock* object with *OBIS* code 0-0:1.0.0.255;
- 1991 • shall be the value recorded by the ESME in attribute 2 of any *Profile generic* object
1992 *entry* as the date-time stamp, at the time the *entry* is added;

- 1993 • shall be the format recorded by the ESME in attribute 2 of any *Profile generic* object
 1994 *entry* in other date-time fields.

1995 Correspondingly:

- 1996 • the '*from_value*' and '*to_value*' fields in the *selective access* structure, which are
 1997 required by the GBCS when accessing attribute 2 of any *Profile generic* object
 1998 directly, shall be *double-long-unsigned* attributes containing a date-time specified in
 1999 seconds since 0 hours 0 minutes 0 seconds on 1st January 2000 UTC; and
- 2000 • the *restricting_object* field in the *selective access* structure shall be set with values of
 2001 class_id = 1; logical_name = 0-0:94.44.100.255; attribute_index = 2 and data_index
 2002 = 0.

2003 The Blue Book requires that, for a Data Protection class object, *restriction_by_date* access
 2004 has *from_date* and *to_date* specified as octet-string. Thus, where a Use Case requires that
 2005 the contents of attribute 2 of a *Profile generic* object are returned in Encrypted form (and so
 2006 accessed via a Data Protection object):

- 2007 • the *from_date* and *to_date* fields in the Command shall be octet-strings formatted as
 2008 per section 4.1.6.1 of the Blue Book; and
- 2009 • the ESME shall undertake the conversion necessary to equate these values to
 2010 '*from_value*' and '*to_value*' equivalents in accessing attribute 2 of any *Profile generic*
 2011 object.

2012 9.1.6 Start of Time and End of Time values

2013 Where a date-time is specified as a 32 bit long unsigned integer:

- 2014 • the Start of Time shall mean the value 0x00000000; and
 2015 • the End of Time shall mean the value 0xFFFFFFFF.

2016 When a date-time is specified in the DLMS COSEM octet-string(12) format:

- 2017 • the Start of Time shall mean the value 0x0000000000000000000000008000FF; and
 2018 • the End of Time shall mean the value 0xFFFFFFFFFFFFFFFFFFFFFFFF8000FF'.

2019 9.1.7 GCS28 Set Clock on GSME

2020 This Use Case covers the setting of time by the Supplier on a GSME.

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Command and Response
Message Type Category	SME.C.C
Capable of future dated invocation?	No
Protection Against Replay Required?	Yes
SEC User Interface Services Schedule (Service Request) Reference	6.11
Valid Target Device(s)	GSME
Valid Business Originator role(s) for Command invocation (and so, for DLMS COSEM Commands, which Application Association is to be used for delivery of the Command to the Device) [Remote Party Messages Only]	Supplier

Valid Response Recipient role(s) (only for Messages authorised by the Access Control Broker on behalf of parties not known to the Device) [Remote Party Messages Only]	N/A
Valid initiating Device type(s) [HAN Only Messages]	N/A
Protocol	ASN.1

2021 Table 9.1.7: Use Case Cross References for GCS28 Set Clock on GSME

2022 **9.1.7.1 Pre-conditions**

2023 None.

2024 **9.1.7.2 Construction of Command**

2025 Set Clock Command Payloads shall be constructed according to the requirements of Section
2026 9.1.7.4 and populated as specified in Table 9.1.7.2.

2027 MAC Header, Grouping Header, KRP Signature and ACB-SMD MAC shall be populated as
2028 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

2029 Table 9.1.7.2: @SetTime.CommandPayload population

2030 **9.1.7.3 Device processing of Command and Response handling**

2031 The GSME receiving a Set Clock Command shall undertake processing steps in the
2032 sequence defined in this Section 9.1.7.3.

2033 The GSME shall:

- 2034 1. undertake Command Authenticity and Integrity Verification as required for a Command
2035 of the SME.C.C Message Category;
- 2036 2. request the now current Communications Hub Time. If the Communications Hub cannot
2037 supply a valid time, it shall provide 0xFFFFFFFF. If this is sent, or no response is
2038 received, the GSME shall:
 - 2039 a. if its current Time Status is set to 'Reliable', set Time Status to 'Unreliable', set
2040 deviceTimeStatus to unreliable, populate deviceTime with its current Time
2041 and process from step 5; or
 - 2042 b. if its current Time Status is not set to 'Reliable', set deviceTimeStatus to be Time
2043 Status, populate deviceTime with its current Time and process from step 5.
- 2044 3. if ((the Communications Hub Time < validityIntervalStart) or (Communications
2045 Hub Time > validityIntervalEnd)), set Time Status to 'Unreliable', and set
2046 deviceTimeStatus to unreliable, leave its Time unchanged, populate
2047 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:

- 2146 • a 'cancellation' can be effected by sending a new Command where the activation
2147 date / time in the new Command has a value that means 'never' to the Device; and
- 2148 • a 'modification' can be effected by sending a new Command where the activation
2149 date-time and / or the 'next' values in the new Command are different than the old
2150 one.

2151 Commands that are marked 'Capable of future dated invocation?' in the Mapping Table can
2152 also be invoked immediately, as specified in Section 9.2.2.4.

2153 When there is a change of Supplier on a Device which is (1) after a future dated change is
2154 stored but (2) before it is activated, the processing at Section 13.3.5.10 will be undertaken at
2155 the point of update of Security Credentials. This ensures future dated commands from the
2156 old Supplier will not be actioned by the Device.

2157 **9.2.2.2 Date-times in future datable commands**

2158 Where a Command contains more than one activation date-time field, the values in all
2159 activation date-times in an instance of that Command shall be the same, except for ZSE
2160 Command Payloads where a field contains 0xFFFFFFFF or 0xFFFFFFFEE, then all activation
2161 date-times shall be either 0xFFFFFFFF or 0xFFFFFFFEE. Devices shall reject Commands
2162 not complying with this requirement. A future datable Command shall be future dated when
2163 it contains an activation date-time that is not one of the values in Section 9.2.2.4.

2164 **9.2.2.3 Effect on prior Commands of the same Message Code**

2165 On receipt of an Authenticated future datable Command, the Device shall overwrite all parts
2166 of any previously sent future dated Command of the same Message Code and, if the
2167 activation date-times for the instructions in the Command are in the past, the Device shall
2168 execute the instructions immediately.

2169 **9.2.2.4 Using a future dated Command to write Attributes immediately**

2170 Where a Command is marked 'Capable of future dated invocation?' in the Mapping Table,
2171 instructions within the Command shall be executed immediately after Authentication by the
2172 Device when:

- 2173 • for DLMS COSEM Commands Payloads, activation date-time(s) have the value
2174 0x000000000000000008000FF;
- 2175 • for ZSE Command Payloads, the activation date-time(s) have the value 0x00000000;
2176 and
- 2177 • for ASN.1 Command Payloads, the activation date-time is not present.

2178 **9.2.2.5 Cancellation of future dated Commands for the writing of Attributes**

2179 Where a Command is marked 'Capable of future dated invocation?' in the Mapping Table,
2180 instructions within the Command shall never be executed by the Device when:

- 2181 • for DLMS COSEM Commands Payloads, activation date-time(s) have the value
2182 0xFFFFFFFFFFFFFFFFFFFF8000FF;
- 2183 • for ZSE Command Payloads, the activation date-time(s) have the value 0xFFFFFFFF
2184 for any ZSE command other than PublishCalendar, PublishSpecialDays,
2185 PublishBlockThresholds, PublishPriceMatrix;
- 2186 • for ZSE Command Payloads, the activation date-time(s) have the value 0xFFFFFFFEE
2187 for the ZSE command PublishCalendar, PublishSpecialDays,
2188 PublishBlockThresholds, PublishPriceMatrix; and
- 2189 • for ASN.1 commands, the activation date-time has a value of 99991231235959Z.

2190 For clarity, sending such a Command has the effect of ‘cancelling’ any previously sent future
2191 datable Command of the same Message Code that the Device has not already executed.

2192 **9.2.2.6 Reactions to Future Dated Commands**

2193 Subject to Command Authenticity and Integrity Verification as detailed in Section 6, where a
2194 Command is future dated, at time the Command is received, the Device shall send a
2195 Response to the Command:

- 2196 • where activation date-times are in the past or the instructions detail immediate
2197 execution as per Section 9.2.2.4, the Command shall be executed immediately, the
2198 Response shall detail the outcome of the Command’s execution, and no Alert shall
2199 be generated;
- 2200 • where activation date-times are in the future, that Response shall detail the success
2201 or otherwise of storing the details in the Command; and
- 2202 • if the activation date-times are in the future and the Command’s details were
2203 successfully stored, the Device shall, at the time the future activation date-time is
2204 reached, process each of the instructions, which contain an activation date-time, as
2205 specified in the Command in the sequence specified in that Command and then
2206 generate an Alert with an Alert Payload, for each instruction, of the same type as the
2207 Payload type of the Command and an Alert Code of 0x8F66 for successful execution
2208 and 0x8F67 for failed execution. Thus:
 - 2209 ○ an ASN.1 Command Payload shall lead to an ASN.1 Alert Payload, which shall be
2210 as defined in Sections 11 and 13;
 - 2211 ○ a DLMS COSEM Command Payload shall lead to DLMS COSEM Alert
2212 Payload(s), which shall be as defined in Table 7.2.9c, where the Use Case
2213 specific additional content contains the concatenation 0x09 || 0x13 || Message
2214 Code || Originator Counter || cosem-attribute-descriptor from the corresponding
2215 part of the Command Payload. Note that 0x09 is the DLMS COSEM tag for octet-
2216 string and 0x13 is the length of the concatenation Message Code || Originator
2217 Counter || cosem-attribute-descriptor; or
 - 2218 ○ a GBZ Command Payload shall lead to GBZ Alert Payload(s), shall be as defined
2219 in Table 7.2.10c, where the Use Case specific additional content contains the
2220 concatenation 0x0E || Message Code || Originator Counter || Extended Header
2221 Cluster ID || Frame control || Command identifier from the corresponding part of
2222 the Command payload.

2223 **9.2.2.7 ESME requirements for activation of future datable Commands**

2224 When either (1) a Command successfully sets the ‘passive’ *Account* object’s
2225 *account_activation_time* (attribute ID 13, OBIS code 0-1:19.0.0.255) to
2226 0x00000000000000000800FF or (2) the ESME’s clock reaches the value in that attribute
2227 or (3) the *activate_account* method is invoked, the ESME shall not activate ‘passive’ *Account*
2228 object, as detailed in the Blue Book, but shall set the attributes in the ‘active’ *Account* object
2229 (OBIS code 0-0:19.0.0.255) as follows:

- 2230 • set the *payment_mode* part of the *account_mode_and_status* attribute (attribute ID 2)
2231 to the *payment_mode* value in the ‘passive’ *Account* object; and
- 2232 • set each of the other *static* attributes (as defined in the Blue Book) to the
2233 corresponding value in the ‘passive’ *Account* object.

2234 When either (1) a Command successfully sets an Activation Date-Time Attribute in Table
2235 9.2.2.7 to 0x00000000000000000800FF or (2) the ESME’s clock reaches the value in an
2236 Activation Date-Time Attribute in Table 9.2.2.7, the ESME shall set the corresponding Active
2237 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	0-0:12.1.1.255	2	10	0-0:12.0.1.255	2
LoadLimitPeriod(Timer)	9000	0-0:94.44.128.6	6	9000	0-0:94.44.128.6	4	71	0-0:17.0.0.255	6
LoadLimitPowerThreshold	9000	0-0:94.44.128.7	6	9000	0-0:94.44.128.7	4	71	0-0:17.0.0.255	4
LoadLimitRestorationPeriod(Timer)	9000	0-0:94.44.128.8	6	9000	0-0:94.44.128.8	4	71	0-0:17.0.0.255	7
AuxiliaryLoadControlSwitchesCalendar	9000	0-0:94.44.128.26	6	10	0-1:12.0.2.255	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
AuxiliaryLoadControlSwitchesCalendar(SpecialDays)	9000	0-0:94.44.128.35	6	11	0-1:11.0.3.255	2	11	0-0:11.0.3.255	2

2238

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 Type 1 devices.

10.2.2 Requirements for the Tunneling Cluster

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

- 2282 Devices supporting the *Tunneling Cluster* as a *Server* shall have a
 2283 *MaximumIncomingTransferSize* set to 1500 octets, in line with the ZSE default. All Devices
 2284 supporting the *Tunneling Cluster* shall use this value in any *RequestTunnelResponse*
 2285 command and any *RequestTunnel* command.
- 2286 Devices shall set the value of the *ManufacturerCode* field in any *RequestTunnel* command
 2287 to 0xFFFF ('not used').
- 2288 The *ProtocolID* of all Remote Party Messages shall be 6. Devices shall set the value of the
 2289 *ProtocolID* field in any *RequestTunnel* command to 6.
- 2290 Devices shall set the value of the *FlowControlSupport* field in any *RequestTunnel* command
 2291 to 'False'.
- 2292 All Devices except Type 2 Devices and GPFs shall support the *Tunneling Cluster* and, within
 2293 that Cluster, the use of the protocol with a *ProtocolID* of 6.
- 2294 An ESME and an HCALCS shall support the *Tunneling Cluster* as a *Server*.
- 2295 A GSME, a PPMID and an HHT shall support the *Tunneling Cluster* as a *Client*.
- 2296 A CHF shall support the *Tunneling Cluster* as a *Client* and as a *Server*.
- 2297 A GPF shall support mirroring functionality. The *Basic Cluster Physical Environment*
 2298 attribute shall be supported and shall have the value 0x01.
- 2299 When a Device receives a *CloseTunnel* command, the Device shall not close that tunnel
 2300 unless the command is sent from the Device which opened the tunnel.
- 2301 Where a Device is required to support the Tunneling Cluster as a server, the server Device
 2302 shall:
- 2303 • when sending a *TransferData* command, use the last *TunnelID* which it holds and
 2304 reset the inactivity counter accordingly. If it receives a *TransferDataError* command in
 2305 response with a *TransferDataStatus* of 0x00 ('No such tunnel'), it shall dis-regard the
 2306 last *TunnelID* and close the associated tunnel;
 - 2307 • persist *TunnelID* values, and the associated client Device's short address and
 2308 endpoint, including when power is lost; and
 - 2309 • when receiving a *RequestTunnel* command from a client Device which has an active
 2310 *TunnelID* registered with the server Device, reset the corresponding inactivity timer
 2311 for that active *TunnelID* and send a *RequestTunnelResponse* containing that
 2312 active *TunnelID*.
- 2313 Where a Device is required to support the Tunneling Cluster as a client, the client Device
 2314 shall:
- 2315 • whenever it sends a *TransferDataError* command with a *TransferDataStatus* of 0x00
 2316 ('No such tunnel') to a server Device, then send a *RequestTunnel* command to that
 2317 server Device; and
 - 2318 • whenever it receives a *TransferDataError* command with a currently valid *TunnelID*
 2319 and a *TransferDataStatus* of 0x00 ('No such tunnel') from a server Device, close any
 2320 associated tunnel and send a *RequestTunnel* to that server Device.
- 2321 **10.2.2.1 Tunneling Requirements**
- 2322 As soon as possible after a Device, which is required to support the *Tunneling Cluster* as a
 2323 client, has successfully established a shared secret key using *CBKE* with a Device, which is
 2324 required to support the *Tunneling Cluster* as a server, the *tunneling* client Device shall send
 2325 a *RequestTunnel* command to the *tunneling* server Device to request a *tunnel* association
 2326 with the Device.

2327 Where such a *tunneling* server Device remains in the *tunneling* client's Device Log, the
 2328 *tunneling* client Device shall send a *RequestTunnel* command to the Device whenever
 2329 0xFFFF seconds have elapsed since receipt of the most recent *RequestTunnelResponse*
 2330 command from that Device.

2331 Where the CHF receives a *RequestTunnelResponse* command from a Device with a
 2332 *TunnelStatus* of 0x01 (*Busy*), the CHF shall send another *RequestTunnel* command three
 2333 minutes later.

2334 Where the CHF receives a *RequestTunnelResponse* command from a Device with a
 2335 *TunnelStatus* of 0x02 (*No More Tunnel IDs*), the CHF shall send a *CloseTunnel* command
 2336 for any *TunnelID* that may relate to an active tunnel association with that Device and, after
 2337 receiving responses to all such commands, send another *RequestTunnel* command.

2338 10.2.2.2 GSME

2339 When a GSME has successfully established a shared secret key using *CBKE* with a
 2340 Communications Hub, the GSME shall:

- 2341 • send a request to the *ZigBee Gas ESI Endpoint* requesting the creation of mirrored
 2342 *Basic*, *Metering* and *Prepayment Clusters* using the *RequestMirror* command;
- 2343 • configure, using the *ConfigureMirror* command, the *ZigBee Gas Mirror Endpoint* to
 2344 use the two way mirroring notification scheme '*Predefined Notification Scheme B*';
 2345 and
- 2346 • send a *RequestTunnel* command to the CHF to request a tunnel association with the
 2347 CHF.

2348 Where the Communications Hub has successfully actioned a *ConfigureMirror* command, the
 2349 GPF shall set the *Push All Static Data - Basic Cluster*, *Push All Static Data - Metering*
 2350 *Cluster* and *Push All Static Data - Prepayment Cluster flags*.

2351 For clarity, the GSME:

- 2352 • shall not action ZSE / ZCL commands received from the GPF in relation to any of the
 2353 flags within *NotificationFlags2*, *NotificationFlags3* and *NotificationFlags5*;
- 2354 • for *NotificationFlags4*, shall only action ZSE / ZCL commands received from the GPF
 2355 in relation to the flags specified in Table 10.2.2.2a.

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

2356 Table 10.2.2.2a: flags in *NotificationFlags4* to be actioned by the GSME

- 2357 • for *FunctionalNotificationFlags*, shall only action ZSE / ZCL commands received from
 2358 the GPF in relation to the flags specified in Table 10.2.2.2b:

Bit Number	Waiting Command
0	<i>New OTA Firmware</i>
1	<i>CBKE Update Request</i>
4	<i>Stay Awake Request HAN</i>
5	<i>Stay Awake Request WAN</i>
6-8	<i>Push Historical Metering Data Attribute Set</i>
9-11	<i>Push Historical Prepayment Data Attribute Set</i>
12	<i>Push All Static Data - Basic Cluster</i>

13	<i>Push All Static Data - Metering Cluster</i>
14	<i>Push All Static Data - Prepayment Cluster</i>
15	<i>NetworkKeyActive</i>
21	<i>Tunnel Message Pending</i>
22	<i>GetSnapshot</i>
23	<i>GetSampledData</i>

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.

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

2396 • where a Message is not being sent (so when the GSME is requesting that a Message
2397 is sent), set the *Data* parameter payload in a *TransferData* command to the value of
2398 Tunnel Manager Header.

2399 A mechanism is also required to notify the GSME that one or more Commands are available
2400 for retrieval from the CHF.

2401 The ZSE specification has a flag called *Tunnel Message Pending* in the *Functional Flag*
2402 *Notification* definition. This flag is used to notify a GSME that the CHF has a Message
2403 waiting to be transferred to the GSME. The flag is set on the first pending Command and is
2404 reset when all Messages have been transferred to the GSME. The flag is available through
2405 the *ReadAttribute* or *MirrorAttributeResponse* command. The requirements for setting this
2406 flag are specified in Section 10.3.4.

2407 The Tunnel Manager Header identifies three different kinds of *TransferData* command
2408 usage:

- 2409 • GET (the value 0x01): this is used by the GSME to retrieve waiting Message from the
2410 CHF;
- 2411 • GET-RESPONSE (the concatenation 0x02 || (Number of Messages remaining for
2412 retrieval after this Message) || (Message addressed to the GSME)): this is used by
2413 the CHF to send a Message to the GSME. It also indicates how many Messages
2414 have yet to be retrieved; and
- 2415 • PUT(the value 0x03): this is used by the GSME to send a Message via the CHF.

2416 Where a Command is waiting on the CHF for the GSME to retrieve it, the following sequence
2417 shall apply:

- 2418 1. the *Tunnel Message Pending* flag is set on the Communications Hub as detailed in
2419 Section 10.3.4;
- 2420 2. the GSME turns on its HAN Interface and obtains the value of the *Tunnel Message*
2421 *Pending* flag; and
- 2422 3. If the *Tunnel Message Pending* flag is set:
 - 2423 a) the GSME sends a *TransferData* command to the CHF with the GET structure
2424 in the Tunnel Manager Header. The Tunnel Manager Header is the only
2425 content in the *Data* field of this *TransferData* command;
 - 2426 b) the CHF sends a *TransferData* command to the GSME with the GET-
2427 RESPONSE structure in the Tunnel Manager Header and a Message in the
2428 remaining part of the *Data* field of the command. The GET-RESPONSE
2429 structure details how many more Messages are available for retrieval; and
 - 2430 c) the GET and GET-RESPONSE pattern repeats until all Messages have been
2431 transferred or the GSME decides to stop requesting Messages.

2432 When the GSME wishes to send a Message, the GSME sends a *TransferData* command to
2433 the CHF with the PUT structure in the Tunnel Manager Header and the Message in the
2434 remainder of the *Data* field in the *TransferData* command.

2435 **10.2.4 TransferData commands sent between GSME and CHF**

2436 When it wishes to send a Message, so an Alert or Response or GBT Message, a GSME
2437 shall send a *TransferData* command to the CHF with the value in the *Data* parameter
2438 payload of the *TransferData* command set to the concatenation:

2439 0x03 || Message

2440 When it wishes to retrieve a Message stored for it on a CHF, a GSME shall send a
2441 *TransferData* command to the CHF with the value in the *Data* field set to 0x01. When the

2442 CHF receives such a *TransferData* command from a GSME, the CHF shall send a
 2443 *TransferData* command to the GSME with the value in the *Data* parameter payload set to:

2444 • the concatenation

2445 0x02 || (Number of Messages remaining for retrieval after this Message) || (Message
 2446 addressed to the GSME)

2447
 2448 where it has Messages for the GSME not yet downloaded by the GSME; or

2449 • the concatenation 0x02 || 0x00, where it has no Messages for the GSME to retrieve,
 2450 the 0x00 representing the number of Messages available to retrieve.

2451 10.3 GSME and GPF interactions

2452 10.3.1 Introduction – informative

2453 The GSME is informed that Remote Party or SME.C.PPMID-GSME Commands are
 2454 available for it to retrieve via *Tunnel Message Pending* flag on the GPF.

2455 The GSME should, under normal operating circumstances, retrieve all Commands buffered
 2456 for it when it turns its HAN Interface on. For example, if two Commands are buffered for it,
 2457 the GSME should retrieve both Commands before turning its HAN Interface off.

2458 However, in some circumstances, a GSME may choose not to retrieve all buffered
 2459 Commands in a single session. In such cases, the GSME should retrieve each Command
 2460 as soon as possible after that Command is received by the CHF.

2461 Potential reasons for a GSME failing to retrieve all buffered Commands include:

- 2462 • the GSME battery requires time to recover;
- 2463 • the GSME is entering a ‘low battery’ mode and limiting the use of its radio; or
- 2464 • a radio communications error.

2465 Section 10.3 details actions the CHF may take where Commands, or GBT Messages
 2466 containing parts of Commands, for a GSME are not retrieved by the GSME.

2467 Commands addressed to a GSME must be processed by the GSME and, when successfully
 2468 processed, any changed operational or configuration data must be made available to the
 2469 GPF. The GPF then has updated information to provide to other Devices on the same
 2470 SMHAN.

2471 In ZSE terms, the GPF incorporates two distinct logical Devices, which are discoverable and
 2472 addressed on different *endpoints*. Section 7 describes which *clusters* reside on which
 2473 *endpoint*.

2474 10.3.2 GSME data residing on the *ZigBee Gas Mirror Endpoint* – 2475 informative

2476 The *ZigBee Gas Mirror Endpoint* provides a ‘reflection’ of the data held by the GSME. A
 2477 GSME is typically a battery-powered Device and its HAN Interface is mostly not turned on,
 2478 making it unable to respond to other Devices. The GSME turns its HAN Interface on at
 2479 regular intervals (e.g. 30 minutes) and pushes consumption data to the *ZigBee Gas Mirror*
 2480 *Endpoint*. This provides other Devices on the same SMHAN with access to GSME
 2481 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);
- 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:

- 2523 1. the CHF shall buffer the Command and instruct the GPF to set the *Tunnel Message*
 2524 *Pending* flag to inform the GSME that the Command is awaiting retrieval. If the
 2525 Command has been sent as multiple GBT Messages, the GPF *Tunnel Message*
 2526 *Pending* flag shall only be set once all GBT Messages making up the Command have
 2527 been received by the Communications Hub. If not all GBT Messages making up a
 2528 Command have been received by a Communications Hub within 24 hours of the first
 2529 GBT Message in that Command being received, then the CHF may discard the GBT
 2530 Messages that have been received for that command;
- 2531 2. if 24 hours elapse after setting the GPF *Tunnel Message Pending* flag without the
 2532 Command being retrieved by the GSME, the CHF may discard the Command. If the
 2533 CHF discards a Command in this way, it shall notify the GPF and the GPF shall log the
 2534 event in its Event Log and send an Alert with a GBZ Payload containing an Alert Code
 2535 0x819D;
- 2536 3. when the GSME turns its HAN Interface on, it shall read the *Tunnel Message Pending*
 2537 flag and retrieve the Command using the *TransferData* command as defined in Section
 2538 10.2.3. Each *TransferData* command received by the GSME shall result in the GSME
 2539 sending a *DefaultResponse* command;
- 2540 4. the CHF shall process the *DefaultResponse* commands it receives to establish when the
 2541 Command has successfully been retrieved by the GSME, and shall provide an indication
 2542 to the GPF accordingly. The GPF shall, when there are no further Commands or GBT
 2543 Messages pending retrieval by the GSME, clear the *Tunnel Message Pending* flag;
- 2544 5. if a Command is a TOM Command, the CHF shall retain a copy of the Command
 2545 contents. For each such Command, the CHF shall start a response timer at the point
 2546 where it has received *DefaultResponse* command(s) confirming the GSME has
 2547 successfully retrieved the Command;
- 2548 6. once a Command is successfully retrieved by the GSME, the GSME shall process the
 2549 Command in line with the requirements of the GBCS. Note that (1) this processing shall
 2550 result in the GSME attempting to send a Response to the Command or an Alert that it
 2551 has received an invalid Command and (2) if sending a Response, the Response shall,
 2552 as per the GBCS requirements, detail the success or failure of GSME processing for
 2553 each instruction within the corresponding Command;
- 2554 7. the GSME shall not, under normal operating conditions, delay sending the Response
 2555 and shall, where possible, send it before turning its HAN Interface off;
- 2556 8. on receipt of a Response that is a TOM Response, the CHF shall inspect the Response
 2557 from the GSME. If the Response indicates successful execution of at least one
 2558 elemental ZCL / ZSE command in the corresponding TOM Command, the CHF shall
 2559 transfer a copy of the corresponding TOM Command contents and the TOM Response
 2560 to the GPF, and shall clear the response timer for the Command;
- 2561 9. on receipt of a TOM Response and the corresponding TOM Command contents, the
 2562 GPF shall clear any stored copy it has of a TOM Command and then:
 - 2563 ○ if the TOM Command is not future dated, process the elemental ZCL / ZSE
 2564 commands contained within the Command according to the *status* within the
 2565 Response, updating data it holds accordingly. Once processed by the GPF, the
 2566 GPF shall make any updated data available over the WAN and over the HAN to
 2567 the Devices in the GPF's Device Log;
 - 2568 ○ if the TOM Command is future dated, store a copy of the TOM Command without
 2569 updating any data it makes available over the WAN or HAN;
- 2570 10. if a Response to a TOM Command has not been received by the Communications Hub
 2571 when the corresponding response timer reaches 6 hours:

- 2572 ○ the CHF may discard its copy of the TOM Command contents, clear the response
2573 timer and notify the GPF accordingly; and
- 2574 ○ on receipt of such a notification, the GPF shall log the event in its Event Log and
2575 send an Alert with a GBZ Payload containing an Alert Code 0x819E;
- 2576 11. for clarity, the CHF shall relay all Remote Party Responses received on its HAN
2577 Interface through the WAN interface;
- 2578 12. whenever the CHF receives an Alert detailing activation of a future dated ZCL / ZSE
2579 command from within a TOM Command (so an Alert with Alert Code 0x8F66 where the
2580 Message Code in the Alert Payload is 0x006B or 0x00A3), the CHF shall pass a copy of
2581 that Alert to the GPF;
- 2582 13. on receipt of such an Alert the GPF shall compare the Originator Counter in the Alert
2583 Payload with the Originator Counter of any copy of a TOM Command it holds with the
2584 same Message Code as in the Alert Payload, and:
- 2585 ○ if the Originator Counters match, the GPF shall update the data it shares over the
2586 HAN and WAN with the elemental ZCL / ZSE command contained within the TOM
2587 Command; or
- 2588 ○ if the Originator Counters do not match or the GPF does not hold a TOM
2589 Command with this Message Code, the GPF shall send an Alert with Alert Code
2590 0x819E, as specified by Section 16.

2591 10.4 GPF Structured Data Items

2592 Underlined terms in this Section 10.4 shall have their meaning in the SMETS and / or CHTS.

2593 10.4.1 Introduction – informative

2594 There are GPF requirements to store structured data items which do not have a direct one to
2595 one mapping in ZSE, or the interpretation may be uncertain. These structured data items
2596 have to be constructed by the GPF.

2597 10.4.2 Structured Data Items

2598 This Section 10.4.2 details how each structured data item shall be constructed by the GPF.

2599 10.4.2.1 Daily Read Log

2600 The GSME shall record the Daily Read Log data items at midnight UTC as defined in
2601 SMETS.

2602 The GSME shall use the *snapshot cause* 'General' (0x0001) for the *snapshot* taken.

2603 The GSME shall push the *snapshot* to the GPF using the *PublishSnapshot* command with a
2604 *SnapshotPayloadType* and a *SnapshotSub-Payload* populated in line with the requirements
2605 of Use Case 'GCS16a Read GSME Daily Read log(s)'. It is not necessary for the GSME to
2606 report any attributes which duplicate those contained in the *snapshot*.

2607 The GPF shall populate the relevant attributes upon receipt of the *PublishSnapshot*
2608 command, providing the command is received between midnight (UTC) and the next
2609 scheduled wake of the GSME.

2610 The GPF shall store the data contained in the *PublishSnapshot* command in the GPF copy
2611 of the GSME Daily Read Log.

2612 In the event of a communications outage, the GPF shall retrieve missing *snapshots* using
2613 the *GetSnapshot* command, with the UTC start time field populated based on the last
2614 received *snapshot* timestamp, if one has been received.

2615 **10.4.2.2 Prepayment Daily Read Log**

2616 If the GSME is operating in prepayment mode it shall record the Prepayment Daily Read Log
 2617 data items at midnight UTC. In ZSE terms, the GSME shall take a *prepayment snapshot* of
 2618 the relevant items. The format and data of the *prepayment snapshot* taken is defined in
 2619 ZSE.

2620 The GSME shall use the *snapshot cause* 'General' (0x0001) for the *prepayment snapshot*
 2621 taken.

2622 The GSME shall push the *prepayment snapshot* to the GPF using the *Publish Prepay*
 2623 *Snapshot* command.

2624 The GPF shall populate the relevant attributes upon receipt of the *Publish Prepay Snapshot*
 2625 command, providing the command is received between midnight (UTC) and the next
 2626 scheduled wake of the GSME.

2627 The GPF shall store the data contained in the *Publish Prepay Snapshot* command in the
 2628 GPF copy of the GSME Prepayment Daily Read Log.

2629 In the event of a communications outage, the GPF shall retrieve missing *prepayment*
 2630 *snapshots* using the *GetPrepaySnapshot* command (and *GetPrepaySnapshot* notification
 2631 flag) with the UTC start time field populated based on the last received *prepayment snapshot*
 2632 timestamp, if one has been received.

2633 **10.4.2.3 Billing Data Log – informative**

2634 SMETS defines Billing Data Log as a log capable of storing the following UTC date and time
 2635 stamped entries:

- 2636 • twelve entries comprising Tariff TOU Register Matrix, the Consumption Register and
 2637 Tariff Block Counter Matrix;
- 2638 • five entries comprising the value of prepayment credits;
- 2639 • ten entries comprising the value of payment-based debt payments; and
- 2640 • twelve entries comprising Meter Balance, Emergency Credit Balance, Accumulated
 2641 Debt Register, Payment Debt Register and Time Debt Registers [1 ... 2].

2642 Requirements for each part are detailed separately in the following sections.

2643 **10.4.2.4 Billing Data Log – Tariff TOU Register Matrix, the Consumption Register and** 2644 **Tariff Block Counter Matrix**

2645 The GSME shall capture this *snapshot* at the following trigger points:

- 2646 • End of Billing Cycle (snapshot cause 'End of Billing Period');
- 2647 • Change of Payment Mode (snapshot cause 'Change of Meter Mode');
- 2648 • Change of Tariff (snapshot cause 'Change of Tariff Information'); and
- 2649 • as specified in Section 13.3.5.10 (snapshot cause 'Change of Supplier').

2650 When it next turns on its HAN Interface, the GSME shall push this *snapshot* to the GPF
 2651 using the *PublishSnapshot* Command with a *SnapshotPayloadType* and a *SnapshotSub-*
 2652 *Payload* populated in line with the requirements of Use Case 'GCS15b Read GSME Billing
 2653 Data Log (change of mode / tariff triggered)'.

2654 The GPF shall store the data contained in the *PublishSnapshot* command in the GPF copy
 2655 of the GSME Billing data Log.

2656 In the event of a communications outage, the GPF shall retrieve missing *snapshots* using
 2657 the *GetSnapshot* command (and the relevant notification flag) with the UTC start time field

2658 populated based on the last received *snapshot* timestamp, if one has been received, or
 2659 0x0000 otherwise.

2660 **10.4.2.5 Billing Data Log – value of prepayment credits**

2661 Upon completion of processing of a valid prepayment top-up, the GSME shall push the latest
 2662 five prepayment top-ups to the GPF using the *PublishTop Up Log* command.

2663 The GPF shall store the data contained in the *Publish Top Up Log* command in the GPF
 2664 copy of the GSME Billing data Log.

2665 If there has been a communications outage, the GPF shall use the *Get Top Up Log*
 2666 command to retrieve all prepayment top-ups that may have been processed during the
 2667 communications outage. The GSME shall set the *Date / Time* field of the *Get Top Up Log*
 2668 command to the current UTC time.

2669 **10.4.2.6 Billing Data Log – payment-based debt payments**

2670 Upon completion of processing of a valid prepayment top-up where the GSME has made a
 2671 debt payment using part of that top-up, the GSME shall push details of that debt payment
 2672 only to the GPF using the *Publish Debt Log* command.

2673 The GPF shall record the details provided in the GPF copy of the GSME Billing Data Log.

2674 In cases of communications outages, the GPF shall request any outstanding payment-based
 2675 debt payments by use of the *GetDebtRepaymentLog* command (and
 2676 *GetDebtRepaymentLog* notification flag) with the Debt Type field set to 0x02 (Debt 3).

2677 **10.4.2.7 Billing Data Log – Meter Balance, Emergency Credit Balance, Accumulated 2678 Debt Register, Payment Debt Register and Time Debt Registers [1 ... 2]**

2679 The GSME shall capture this snapshot at the following trigger points:

- 2680 • End of Billing Cycle (snapshot cause bit 1 set: 'End of Billing Period' , as per
 2681 *PublishSnapshot* command);
- 2682 • Change of Payment Mode (snapshot cause bit 14 set: 'Change of Meter Mode');
- 2683 • Change of Tariff (snapshot cause at least one of the bits set: bit 3 'Change of Tariff
 2684 Information' and / or bit 4 'Change of Price Matrix' and / or bit 5 'Change of Block
 2685 Thresholds'); and
- 2686 • as specified in Section 13.3.5.10 (snapshot cause 'Change of Supplier').

2687 When it next turns on its HAN Interface, the GSME shall push this *snapshot* to the GPF
 2688 using the *Publish Prepay Snapshot* command.

2689 The GPF shall store the data contained in the *Publish Prepay Snapshot* command in the
 2690 GPF copy of the GSME Billing Data Log.

2691 In the event of a communications outage, the GPF shall retrieve missing *snapshots* using
 2692 the *GetPrepaySnapshot* command (and *GetPrepaySnapshot* notification flag) with the UTC
 2693 start time field populated based on the last received snapshot timestamp, if one has been
 2694 received.

2695 **10.4.2.8 GPF Profile Data Log**

2696 The GPF shall create the GPF Profile Data Log from the consumption information pushed by
 2697 the GSME each half hour.

2698 The GSME shall, on each half hour, record the following information and push to the GPF:

- 2699 • the *CurrentSummationDelivered* attribute containing total consumption value (with
 2700 units of m³);

- 2701 • the *CurrentDayAlternative ConsumptionDelivered* attribute containing total
2702 consumption today (with units of kWh); and
- 2703 • the *CurrentDayCostConsumptionDelivered* attribute containing total cost of
2704 consumption today (with units of Currency Unit);
- 2705 Upon receipt of the pushed data, the GPF shall calculate the consumption with units of m³
2706 over the previous half hour by subtracting its previously recorded total consumption value
2707 from the total consumption value now sent.
- 2708 The resulting value shall be stored in the GPF Profile Data Log.
- 2709 In the event that there are missing values in the GPF Profile Data Log, the GPF shall
2710 interrogate the GSME Profile Data Log using the *GetSampledData (SampleID 0x0000)*
2711 command and the *GetSampledData* notification flag to retrieve missing values.
- 2712 **10.4.2.9 GPF Daily Gas Consumption Log**
- 2713 The GPF shall create the GPF Daily Gas Consumption Log based on the values pushed
2714 from the GSME. The difference between last total consumption value pushed from the
2715 GSME each UTC day and the last value pushed in the prior UTC day shall be time stamped
2716 and stored in the GPF Daily Gas Consumption Log, so that the values in the log represent
2717 consumption in that UTC day.
- 2718 In the event of communications outages resulting in the final daily value being missed, the
2719 GPF shall retrieve the values from the GSME Profile Data Log using the *GetSampledData*
2720 (*SampleID 0x0000*) command and *GetSampledData* notification flag.
- 2721 **10.4.2.10 Historical Attributes**
- 2722 A GSME shall support:
- 2723 • the *Historical Cost Consumption Information* attribute set, measured in Currency
2724 Units; and
- 2725 • the *Alternative Historical Consumption* attribute set, measured in kWh.
- 2726 A GPF shall mirror the attribute sets listed above.
- 2727 As per Section 10.4.2.8, the GSME shall, on each half hour, record the following information
2728 and push to the GPF:
- 2729 • total consumption value (with units of m³);
- 2730 • total consumption today (with units of kWh); and
- 2731 • total cost of consumption today (with units of Currency Unit);
- 2732 Using the 'total consumption today' value, the GPF shall update the attributes of the mirrored
2733 *Alternative Historical Consumption* attribute set.
- 2734 Using the 'total cost of consumption today' value, the GPF shall update the attributes of the
2735 mirrored *Historical Cost Consumption Information* attribute set.
- 2736 In exception circumstances, the GPF shall request the GSME to push the historical data sets
2737 using the '*Push Historical Metering Data Attribute Set*' and '*Push Historical Prepayment Data*
2738 *Attribute Set*' notification flags. The GSME shall interpret the '*Push Historical Metering Data*
2739 *Attribute Set*' notification flag as requiring it to push the *Alternative Historical Consumption*
2740 attribute set.
- 2741 **10.4.2.11 Other attributes**
- 2742 The GSME shall populate the *AccumulatedDebt* attribute in line with the SMETS
2743 Accumulated Debt requirements, and all other Devices shall interpret that attribute
2744 correspondingly.

- 2745 The GSME shall populate the *Credit Remaining* attribute in line with the SMETS Meter
 2746 Balance requirements, and all other Devices shall interpret that attribute correspondingly.
 2747 The GSME shall apply functionality related to the *CutOffValue* attribute in line with this
 2748 interpretation of *Credit Remaining* and the SMETS requirements for Disablement Threshold.
- 2749 The GPF shall calculate the value of the price in any ZCL *PublishPrice* command it creates
 2750 using the tariff information it has derived through the TOM and the time from the
 2751 Communications Hub's clock.
- 2752 The ESME shall populate the *AuxSwitchNLabel* attribute in line with the SMETS Auxiliary
 2753 Load Control Switch [n] Description requirements, and all other Devices shall interpret that
 2754 attribute correspondingly.
- 2755 The *CommodityType* and *MeteringDeviceType* attributes shall be set by devices as follows:
- 2756 • 'GPF: Metering Device (Gas Mirror Endpoint)': 128 (Mirrored Gas Metering);
 - 2757 • 'GSME: Metering Device': 1 (Gas Metering);
 - 2758 • 'ESME: Energy Services Interface (Electricity ESI Endpoint)' and not a Polyphase
 2759 ESME: 0 (Electric Metering);
 - 2760 • 'ESME: Energy Services Interface (Electricity ESI Endpoint)' and a Polyphase ESME:
 2761 15 (Electric Metering Element/Phase 3);
 - 2762 • 'ESME: Energy Services Interface (Twin ESME aggregate ESI Endpoint)': 0 (Electric
 2763 Metering);
 - 2764 • 'ESME: Energy Services Interface (Twin ESME primary ESI Endpoint)':13 (Electric
 2765 Metering Element/Phase 1); and
 - 2766 • 'ESME: Energy Services Interface (Twin ESME secondary ESI Endpoint)':14 (Electric
 2767 Metering Element/Phase 2).
- 2768 When processing a ZSE *Get Event Log* command or a ZSE *Clear Event Log* command with
 2769 a Log ID nibble of 0x6 (GSME Event Log) or 0x7 (GSME Security Log), a GPF shall process
 2770 the command using the relevant GSME Proxy Log copy of the GSME Event or Security Log.
 2771 Other values of Log ID shall refer to the GPF's own logs.
- 2772 Where an ESME is not a Twin Element ESME it shall populate the *SiteID* attribute with the
 2773 13 most significant octets being the Import MPAN and the following 13 octets the Export
 2774 MPAN.
- 2775 Where an ESME is a Twin Element ESME it shall populate:
- 2776 • the *SiteID* attribute in the 'ESME: Energy Services Interface (Twin ESME aggregate
 2777 ESI Endpoint)' with the 13 most significant octets being the Import MPAN on the
 2778 primary element and the following 13 octets the Export MPAN; and
 - 2779 • the *SiteID* attribute in the 'ESME: Energy Services Interface (Twin ESME secondary
 2780 ESI Endpoint)' with the most significant 13 octets being the Import MPAN on the
 2781 secondary element.
- 2782 Where a ZCL / ZSE command containing *IssuerEventId* and / or *ProviderID* fields is received
 2783 by a Device as part of a GBZ Remote Party Command, the Device shall undertake no
 2784 processing in relation to those two fields. For clarity, this means the Device shall not use the
 2785 contents of those fields for anti replay purposes.
- 2786 ESME shall support *StartRandomizedMinutes* (identifier 0x00FE) and
 2787 *EndRandomizedMinutes* (identifier 0x00FF) attributes on the *Demand Response and Load*
 2788 *Control Cluster* as a *Server*.
- 2789 In ZSE *GetSampledData* and *GetSampledDataResponse* commands:

- 2790 • the *SampleID* field shall be interpreted as:
 - 2791 ○ 0x0000 meaning Profile Data Log;
 - 2792 ○ 0x0001 meaning Daily Consumption Log; and
 - 2793 ○ 0x0002 meaning Network Data Log; and
- 2794 • the *SampleRequestInterval* field shall contain 0xFFFF whenever the *SampleID* field
2795 is 0x0001.
- 2796 A GSME shall reject any *PublishPriceMatrix* command that does not contain four *Price*
2797 fields.
- 2798 When processing a *Get Snapshot* or *Get Prepay Snapshot* command, a Device shall return
2799 all snapshots where the *Snapshot Cause* in the snapshot matches any of the set bits in the
2800 *Snapshot Cause* parameter of the command. When processing a command where *Issuer*
2801 *Calendar ID* has the value 0xFFFFFFFF or 0xFFFFFFF, a GPF or GSME shall interpret
2802 0xFFFFFFFF as meaning the currently in force Tariff Switching Table calendar and
2803 0xFFFFFFF as meaning the currently in force Non-Disablement Calendar.
- 2804 Devices shall support the requirements relevant to their device type detailed in 'Trust Center
2805 Swap-Out' process section of the ZSE specification.
- 2806 The GSME shall set the *BillDeliveredTrailingDigit* attribute to the same value as
2807 *PriceTrailingDigit* in the *Price* cluster.
- 2808 In line with the SMETS requirement, the *UnitOfMeasure* parameter in the
2809 *PublishTariffInformation* command, sent to a GSME shall be 0x00 (kWh) as per the Message
2810 Templates for GCS01a and GCS01b, shall apply to the *Block Threshold N* parameter in the
2811 *PublishBlockThresholds* command in such Messages. Contrary to ZSE, the GSME shall
2812 undertake the necessary calculation when comparing these thresholds against the
2813 *CurrentBlockPeriodConsumptionDelivered* attribute (whose unit of measure in line with
2814 SMETS shall be m³).
- 2815 Contrary to ZSE, the GPF shall accept any valid UTCTime in the value of the
2816 *Implementation Date/Time* parameter in a *Publish Change Of Tenancy* command, in a
2817 Command complying with Use Case GCS09.
- 2818 The GSME shall only action a ZSE *Set Low Credit Warning Level* command it receives at
2819 the specified 'From Date Time' specified with that ZSE command.
- 2820 In GBZ Remote Party Commands containing the *Issuer Tariff ID* parameter, the value of that
2821 parameter shall always be 0x00000001.
- 2822 In GBZ Remote Party Commands containing the *Friendly Credit Calendar ID* parameter, the
2823 value of that parameter shall always be 0x00000002.
- 2824 In responding to GBZ Remote Party Commands requesting information relating to the *Price*
2825 *Cluster*, the GSME & GPF shall provide that information in the GBZ Remote Party
2826 Response.
- 2827 On receipt of a *Credit Adjustment* command with a *Credit Adjustment Type* set to 0x01, the
2828 GSME shall reset the prepayment mode Meter Balance, the Emergency Credit Balance and
2829 the Accumulated Debt Register.
- 2830 Where a Device receives a GBZ Command containing a ZSE *Change Payment Mode*
2831 command or a ZSE *Get Current Price* command and the Device fails successfully to execute
2832 that ZSE command, the Device shall populate the corresponding GBZ Response with a ZSE
2833 *Default Response* command in the place of either a ZSE *Change Payment Mode Response*
2834 command or a ZSE *Publish Price* command.

2835 For GSME, the value of the *Multiplier* attribute in the *Metering* cluster shall be set to 1 and
 2836 the value of the *Divisor* attribute in the *Metering* cluster shall be set to 1000.

2837 **10.5 Hand Held Terminal (HHT) interactions**

2838 **10.5.1 Introduction – informative**

2839 An HHT allows for delivery of Remote Party Messages to and from the SMHAN. This is as
 2840 an alternative delivery route to the Communications Hub's WAN connection. It is intended
 2841 for one-off configuration of Devices, for example at installation. Hence, there are time outs
 2842 to ensure usage is limited in this way.

2843 This Section 10.5 specifies requirements related to:

- 2844 • how a connection is made between an HHT and a Communications Hub; and
- 2845 • how Remote Party Messages are then to be transferred to and from the HHT.

2846 **10.5.2 Establishing a connection between an HHT and a** 2847 **Communications Hub – informative**

2848 ZigBee supports an Inter-PAN mechanism which can be used for communication links
 2849 between Devices. This mechanism is used to establish a link key between the HHT and the
 2850 Communications Hub (referred to as the Inter-PAN Link Key (IPLK)), by using ZSE's *CBKE*
 2851 (for clarity, this IPLK is not used to encrypt any Inter-PAN communication). The Inter-PAN
 2852 link is then used for HHT to Communication Hub communication:

- 2853 • the HHT uses the link to send its Entity Identifier and *Install Code* to the
 2854 Communications Hub as part of a 'CCS01 Add Device to CHF device log' Command;
- 2855 • the Communications Hub adds these details to the CHF's Device Log (so allowing
 2856 the HHT to *join* the SMHAN) and confirms this to the HHT using a 'CCS01 Add
 2857 Device to CHF device log' Response; and
- 2858 • the HHT then *joins* the SMHAN and so can exchange Remote Party Messages within
 2859 the Communications Hub, and the Communications Hub can relay them to / from the
 2860 specified Device(s) on the HAN. The IPLK is used as the *pre-configured link key* in
 2861 this *joining* process.

2862 Both the *Inter-PAN communications* and *joining* to the SMHAN use the *CBKE* mechanism
 2863 that is defined in ZSE.

2864 *Inter-PAN* is only to be available for 60 minutes from power on of the Communications Hub.
 2865 So, if needed, *Inter-PAN* can be enabled by power cycling the Communications Hub.

2866 The *Inter-PAN* mechanism defined by ZSE requires the HHT to specify the Communications
 2867 Hub that it wishes to link to. There may be multiple Communications Hubs available to the
 2868 HHT to connect to via *Inter-PAN*.

2869 There are a number of options to provide the HHT with information sufficient to identify
 2870 uniquely the Communications Hub it is to link to, including:

- 2871 • the installer manually reading the GPF's Entity Identifier (which is the IEEE address
 2872 of the Communications Hub's SMHAN radio) printed on the Hub, and confirming /
 2873 selecting this on the HHT; or
- 2874 • the installer using a scanner on the HHT to read the GPF's Entity Identifier.

2875 Two illustrative connection scenarios are provided in the following two sections

2876 **10.5.2.1 Illustration 1: Installer manually chooses network – informative**

- 2877 1. the Communications Hub opens Inter-PAN communication for 60 minutes after power
 2878 on;

- 2879 2. the HHT is powered on;
- 2880 3. the HHT performs an active scan using the *Beacon Request* mechanism;
- 2881 4. the HHT displays the IEEE addresses returned in the *Beacons* from all neighbouring
- 2882 *PAN Coordinators*. Note that the GBCS requires the *Extended PAN ID* initially to be set
- 2883 to the Communications Hub's HAN Interface's IEEE address. This is the same as the
- 2884 GPF Entity Identifier, which is printed on the Communications Hub in line with CHTS.
- 2885 Note that, if the Communications Hub is a replacement for one previously installed, the
- 2886 Extended PAN ID will not be its GPF Entity Identifier, and the installer will need a
- 2887 different way to establish the correct Extended PAN ID to use;
- 2888 5. the installer (who knows the Consumer's Communications Hub's ZigBee IEEE address
- 2889 as the GPF Entity Identifier is printed on the Communications Hub) picks the desired
- 2890 IEEE address;
- 2891 6. the HHT initiates *Inter-PAN CBKE* with the Communications Hub;
- 2892 7. the Communications Hub responds to the *Inter-PAN CBKE*;
- 2893 8. if *Inter-PAN CBKE* completes successfully, the HHT sends its Install Code and Entity
- 2894 Identifier to the Communications Hub in a CCS01 Command, then
- 2895 9. the Communications Hub adds the HHT to the CHF Device Log and sends a CCS01
- 2896 Response to the HHT accordingly, and then
- 2897 10. the HHT *joins* to the SMHAN;
- 2898 11. otherwise, no link is established.

2899 **10.5.2.2 Illustration 2: HHT uses barcode scan – informative**

- 2900 1. the Communications Hub opens Inter-PAN communication for 60 minutes after power
- 2901 on;
- 2902 2. the HHT is powered on;
- 2903 3. the HHT optically scans the GPF Entity Identifier printed on the target Communications
- 2904 Hub;
- 2905 4. the HHT performs an active scan using the *Beacon Request* mechanism;
- 2906 5. when a Beacon returns an IEEE address equal to the scanned GPF Entity Identifier, the
- 2907 HHT initiates *Inter-PAN communication* with the Communications Hub so identified;
- 2908 6. the Communications Hub responds to the *Inter-PAN CBKE*;
- 2909 7. if *Inter-PAN CBKE* completes successfully:
- 2910 8. the HHT sends its Install Code and Entity Identifier to the Communications Hub in a
- 2911 CCS01 Command, then
- 2912 9. the Communications Hub adds the HHT to the CHF Device Log and sends a CCS01
- 2913 Response to the HHT accordingly, and then
- 2914 10. the HHT *joins* to the SMHAN;
- 2915 11. otherwise, no link is established.

2916 **10.5.3 WAN proxy operation**

2917 **10.5.3.1 Introduction – informative**

2918 The HHT has to be capable of holding Remote Party Messages, to which the appropriate
 2919 Remote Party Message protection has already been applied, and has to be capable of
 2920 exchanging such Messages.

2921 The Communications Hub must therefore be able to maintain two effective ‘WAN’ interfaces;
 2922 the real one via the WAN network interface and a ‘logical WAN’ via the connection to the
 2923 HHT.

2924 **10.5.3.2 WAN Responses**

2925 The Communications Hub shall send any Responses and Alerts through both its WAN
 2926 interface and the link to the HHT, if present. Whilst this may result in apparent unsolicited
 2927 Responses at the Remote Party which have to be dealt with, it ensures the earliest possible
 2928 reconciliation of Commands destined for Smart Metering Equipment.

2929 **10.5.3.3 HHT and CHF – Device Requirements**

2930 Any Device which provided its Entity Identifier and Install Code in a CCS01 Command to the
 2931 Communications Hub via Inter-PAN shall be treated by the CHF as being of type HHT. For
 2932 clarity, the Device Type specified in any such CCS01 Command shall be disregarded by the
 2933 CHF.

2934 As per Section 10.2.2, in interactions between an HHT and a Communications Hub over the
 2935 SMHAN:

- 2936 • the HHT shall support the *Tunneling Cluster* as a *Client*; and
- 2937 • the Communications Hub shall support the *Tunneling Cluster* as a *Server*.

2938 The Communications Hub shall only allow *Inter-PAN communications* for 60 minutes from
 2939 any power on of the Communications Hub. For clarity, this is the period during which an
 2940 HHT can establish a connection, not the period of use of any connection.

2941 At power on, a Communications Hub shall remove any Devices of type HHT from the CHF
 2942 Device Log.

2943 The Communications Hub shall prior to installation, set *nwkExtendedPANId* to be the Entity
 2944 Identifier of the GPF, which is always the Communications Hub’s IEEE address for its HAN
 2945 Interface.

2946 **10.5.3.4 HHT and CHF – establishing communications**

2947 Prior to being able to exchange Messages, the HHT and Communications Hub shall
 2948 undertake the following steps:

- 2949 1. the HHT shall identify the Communications Hub and initiate the *CBKE* process using
 2950 *Inter-PAN communications*, as specified in this Section 10.5.3.4;
- 2951 2. the Communications Hub shall not respond to any such communications if more than 60
 2952 minutes has elapsed since the Communications Hub’s most recent power on, or if there
 2953 is a Device of type HHT already in the CHF’s Device Log. Otherwise, the
 2954 Communications Hub shall respond to the *CBKE* request;
- 2955 3. if *CBKE* does not succeed, processing shall cease. Otherwise, the Communications
 2956 Hub and HHT shall each store the link key established through *CBKE* (referred to as the
 2957 Inter-PAN Link Key (IPLK)) along with a record of the other Device’s Entity Identifier,
 2958 which is its IEEE address. Then processing shall continue from step 4;
- 2959 4. the HHT shall send a ‘CCS01 Add Device to CHF device log’ Command, populated with
 2960 the HHT’s Entity Identifier and *install code*, to the CHF using the mechanism defined in
 2961 Section 10.5.3.5;
- 2962 5. where all parts of the Command are successfully received, the CHF shall first validate
 2963 that the Entity Identifier in the CCS01 Command matches the Entity Identifier it stored
 2964 against the IPLK at step 3. If it does not match, processing shall cease. Otherwise the
 2965 CHF shall process the Command according to the requirements of the ‘CCS01 Add
 2966 Device to CHF device log’ Use Case, and shall send any resulting ‘CCS01 Add Device to

2967 CHF device log' Response to the HHT, via Inter-PAN using the mechanism defined in
 2968 Section 10.5.3.5. For clarity the install code in the CCS01 Command shall not be used
 2969 in the subsequent ZigBee *joining*. Where the CHF sends such a Response, it shall start
 2970 a timer. When that timer reaches 0xFFFF seconds, the CHF shall remove the HHT from
 2971 its Device Log, remove the HHT from the SMHAN and close any open tunnels to the
 2972 HHT;

2973 6. if the 'CCS01 Add Device to CHF device log' Response from the CHF to the HHT states
 2974 that the HHT has been added to the CHF's Device Log, the HHT may attempt to join the
 2975 SMHAN. The Devices shall use the IPLK stored at step 3 as the pre-configured link key
 2976 to secure the joining process. If the *joining* is successful:

2977 a) the HHT shall send a *RequestTunnel* command to the CHF, with contents as per
 2978 Section 10.2.2;

2979 b) the CHF shall process the *RequestTunnel* command and send a
 2980 *RequestTunnelResponse* command in response; and

2981 c) if *TunnelStatus* in the *RequestTunnelResponse* command is not 0x00 ('*success*'),
 2982 processing by the HHT shall cease. Otherwise the HHT and CHF may now
 2983 exchange Messages using the *TransferData* command.

2984 Note that steps 1 to 5 above use *Inter-PAN communications*; the remaining step 6 uses the
 2985 standard ZigBee SMHAN communications.

2986 Once the HHT has *joined* the SMHAN, any Messages received by the CHF from the HHT in
 2987 the *Data* parameter payload of a *TransferData* command, shall be forwarded to the relevant
 2988 Device on the SMHAN as if they were received via the Communications Hub's WAN
 2989 interface.

2990 Whilst the HHT is in the CHF's Device Log and *joined* to the SMHAN, any Responses
 2991 received by the CHF from any SMHAN Device shall be provided to the HHT using the
 2992 *TransferData* command. Such Responses shall also be sent over the Communications
 2993 Hub's WAN interface, if available.

2994 Note that the requirements of Section 7.2.11(Transfer of Large Remote Party Messages)
 2995 apply to Messages exchanged once the HHT has joined the SMHAN. Section 7.2.11
 2996 requirements do not apply to the CCS01 Command and Response exchanged using the
 2997 Inter-PAN mechanism.

2998 Once the HHT usage on the SMHAN is complete, the HHT should send a *CloseTunnel*
 2999 command to the Communications Hub. On receipt of such a *CloseTunnel* command from an
 3000 HHT, the Communications Hub shall process that command as per the ZSE specification
 3001 and shall:

- 3002 • remove the HHT from its Device Log; and
- 3003 • remove the HHT from the SMHAN.

3004 **10.5.3.5 Population of the Inter-PAN ZigBee Frame**

3005 A Device may send a Remote Party Message to another Device on the same SM HAN using
 3006 the ZSE Inter-PAN Frame. In doing so, the sending Device shall:

- 3007 • Take the 'n' most significant octets of the Remote Party Message to create an 'Inter-
 3008 PAN Message Fragment (IPMF)', where 'n' is the maximum number such that the
 3009 aMaxPHYPacketSize (with its IEEE 802.15.4 2003 specification meaning) would not
 3010 be exceeded in the resulting message over the SM HAN. This IPMF shall have an
 3011 index of 0x00;
- 3012 • If there are remaining octets in the Remote Party Message, take the next 'n' most
 3013 significant octets to create IPMF with an index of one greater than the previous IPMF;

- 3014 • Repeat step 2 until there are no remaining octets in the Remote Party Message; and
- 3015 • Set the 'Total Number of IPMF' to be the number of IPMF so created.

3016 A Device receiving a Remote Party Message by this mechanism shall respond to each IPMF
 3017 received with an 'IPMF Response' constructed according to this Section 10.5.3.5. The
 3018 receiving Device shall re-construct the Remote Party Message in a way that is aligned with
 3019 the splitting mechanism defined in this Section 10.5.3.5.

3020 If a response to an Inter-PAN ZigBee Frame is required by this Section 10.5.3.5 but the
 3021 receiving Device has not received such a Response after 60 seconds, the receiving Device
 3022 may treat the transmission of the underlying CCS01 Message as having failed.

3023 When a sending Device is populating the Inter-PAN ZigBee Frame to communicate with a
 3024 receiving Device using Inter-PAN, the sending Device shall:

- 3025 • Populate the Inter-PAN ZigBee NWK Header according to the requirements of Table
 3026 10.5.3.5a;
- 3027 • Populate, in the Inter-PAN ZigBee APS Header, the fields specified in Table 10.5.3.5b,
 3028 according to the requirements of Table 10.5.3.5b; and
- 3029 • Populate the Inter-PAN ZigBee Payload:
- 3030 ○ According to the requirements of Table 10.5.3.5c when the Device is sending an
 3031 IPMF in this Frame; or
- 3032 ○ According to the requirements of Table 10.5.3.5d when the Device is sending an
 3033 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	0b00	2	Unicast

Element	Contents	Length (bits)	Note
(bits 2-3)			
- 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

3034 Table 10.5.3.5b: Inter-PAN ZigBee APS Header (five octets total length)

3035 Note, the Inter-PAN ZigBee Payloads specified in this Section 10.5.3.5 do not contain a *ZCL*
 3036 *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 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)

11 Downloading firmware images to Devices

11.1 Introduction – informative

Compared to other Smart Metering messages, firmware images are large. Further, each image is likely to be applicable to a significant number of Devices. Thus, an end-to-end, unicast Message to each affected Device, with each message containing a copy of the image, is not efficient from a WAN perspective.

This leads to the firmware update process being separated into two stages:

- distribution of the image to end Devices without any activation of that image; and
- a separate and subsequent ‘activation’ Command to each Device.

The Distribute Firmware Command is not a Critical Command (since it does not affect the operating firmware) and does not need to be unicast.

The Activate Firmware Command is a Critical Command and so must be unicast – as it must be digitally signed and be for one, and only one specified Device. Further, the Activate Command must apply to one, and only one, image and that image must have originated from the same party that signs the Activate Firmware Command (that is, the party responsible for that Device). To meet these requirements:

- the Activate Firmware Command is of type SME.C.C and so the Signature and MAC on the Command shall have been verified by the Device prior to the Hash validation (see next bullet); and
- a Device receiving an Activate Firmware Command shall calculate a Hash over the Manufacturer Image it holds and ensure the Hash so calculated matches that in the Activate Firmware Command, before the Device attempts to activate the corresponding Manufacturer Image.

The GBCS does not constrain the mechanisms used by Device manufacturers to ensure that only valid Manufacturer Images are activated on Devices manufactured by them. The GBCS does require that the manufacturer information related to a Manufacturer Image is made available, so that the Upgrade Image and the ZigBee Over-The-Air (OTA) Header can be provided when requesting distribution of an image.

In common with other Messages, the GBCS shall not constrain the mechanisms by which the firmware Messages are transported to the Communications Hub. The GBCS constrains HAN transport mechanisms to those provided by ZSE.

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, SMETS and CHTS are meant in this sense.

Therefore, when a Device reports its ‘installed’ or ‘active’ firmware version it shall respond with either:

- 3082 • the firmware version for its mostly recently successfully activated Manufacturer
3083 Image (with the value in the corresponding OTA Header and so the value in the
3084 Certified Products List (CPL)); or
- 3085 • if it has never successfully activated a new Manufacturer Image, the firmware version
3086 it held on installation (which again would correspond to the CPL).

3087 11.2 Common Requirements

3088 11.2.1 Transport of firmware images

3089 Italicised terms in this Section 11.2.1 shall have the meanings defined in ZigBee Document
3090 09-5264r23.

3091 For ESME and GSME firmware image distribution, the ZigBee Over-The-Air (OTA)
3092 mechanisms shall be used for transport of the image over the HAN. The ESME / GSME
3093 firmware image delivered to the Communications Hub shall comply with ZigBee OTA format
3094 requirements.

3095 Communications Hub firmware images shall not be transported over the HAN and so ZigBee
3096 OTA structures shall not be required.

3097 Every Communications Hub shall be configured to act as the single OTA Server on its HAN.

3098 ESME and GSME shall be configured to act as an OTA Client. The ESME shall use the
3099 '*Image Notify*'¹⁹ *Command* sent by the OTA Server to inform it that a new firmware image is
3100 available. The GSME shall use the notification flags mechanism whereby a flag shall be set
3101 by the OTA Server to inform it that a new firmware image is available when requested.

3102 The Communications Hub shall:

- 3103 • as required by CHTS, have the capability to store one GSME OTA Upgrade Image
3104 and one ESME OTA Upgrade Image; and
- 3105 • overwrite an image with a subsequently delivered image for the same Device type
3106 unless:
 - 3107 ○ the subsequently delivered image has Force Replace = 0x00; and
 - 3108 ○ the Communications Hub has sent at least one *Image Block Response Command*
3109 relating to the already stored image but has not received a corresponding
3110 *Upgrade End Request Command*²⁰.

3111 In such circumstances the Communications Hub shall not overwrite the currently stored
3112 image.

3113 Whenever the Communications Hub's OTA Server issues an *Upgrade End Response*
3114 *Command* to a GSME or ESME pursuant to this GBCS, the *UpgradeTime* parameter shall
3115 have the value 0xFFFFFFFF²¹.

3116 The OTA Server shall not issue *Image Block Response Commands* with WAIT_FOR_DATA
3117 status.

3118 Contrary to section 6.13 of ZigBee Document 09-5264r23, the OTA Client shall not activate
3119 any Firmware except as specified in Use Case CS06.

3120 11.2.2 Construction of Upgrade Image

3121 For an ESME or GSME firmware image, the Authorising Remote Party shall be the Supplier
3122 for the target Device.

¹⁹ See section 6.10.3 of ZigBee Document 09-5264r23

²⁰ As defined in section 6.10 of ZigBee Document 09-5264r23

²¹ As defined in sections 6.10.10 and 6.8.4 of ZigBee Document 09-5264r23

3123 For a Communications Hub firmware image, the Authorising Remote Party shall be the WAN
3124 Provider for the target Device.

3125 Upgrade Image shall be the concatenation:

3126 Manufacturer Image || Force Replace || 0x40 || Authorising Remote Party Signature

3127 where:

- 3128 • Manufacturer Image shall contain the firmware image the Device is to apply and any
3129 manufacturer specific data needed. For clarity, the GBCS shall not constrain the
3130 structure or contents of Manufacturer Image;
- 3131 • Force Replace shall be a single octet where Force Replace = 0x00 shall mean do not
3132 force the replacement of the currently stored image; and
- 3133 • Authorising Remote Party Signature shall be calculated across the Manufacturer
3134 Image using the Authorising Remote Party's Private Digital Signing Key.

3135 11.2.3 Construction of OTA Upgrade Image

3136 OTA Upgrade Image shall be the concatenation:

3137 OTA Header || Upgrade Image

3138 where OTA Header shall be populated according to Table 11.2.3. For clarity, there shall be
3139 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	

3140 Table 11.2.3: Population of the OTA Header

3141 The OTA Header shall uniquely identify a firmware image.

3142 **11.2.4 Construction of Manufacturer Image Hash**

3143 Manufacturer Image Hash shall be a Hash calculated across the whole Manufacturer Image
3144 file that is provided to the Authorising Remote Party.

3145 **11.2.5 Verification of the authenticity of the Upgrade Image**

3146 The Device shall verify Upgrade Image by verifying the Authorising Remote Party Signature
3147 using Manufacturer Image and the Authorising Remote Party's Public Key. For clarity, this
3148 shall be the only ECDSA verification required by the GBCS and this is not the ZSE ECDSA
3149 Signature sub-element.

3150 For an ESME or GSME receiving an Upgrade Image, the Authorising Remote Party's Public
3151 Key shall be that held by the Device in the {supplier, digitalSignature,
3152 management} Trust Anchor Cell.

3153 For a Communications Hub receiving an Upgrade Image, the Authorising Remote Party's
3154 Public Key shall be that held by the Device in the {wanProvider, digitalSignature,
3155 management} Trust Anchor Cell.

3156 **11.2.6 Construction of Firmware Distribution Receipt Alert**

3157 If the Device is an ESME, the 'Alert Payload' fields shall be populated according to Section
3158 7.2.9.

3159 If the Device is a GSME, the 'Alert Payload' fields shall be populated according to Section
3160 7.2.10.

3161 In both cases, the Device shall:

- 3162 • populate the Use Case Specific Additional Content with the concatenation
3163 0x0920 || the calculated Manufacturer Image Hash
- 3164 • populate the Alert Code field with 0x8F1C (failure), or 0x8F72 (success).

3165 **11.2.7 Activation of firmware images**

3166 The Activate Firmware Command shall be of type SME.C.C.

3167 A Device receiving such a Command shall undertake the verifications required of a SME.C.C
3168 Command.

3169 If all such SME.C.C verifications succeed, the Device shall then calculate Manufacturer
3170 Image Hash over the Manufacturer Image it holds and compare that with the Manufacturer
3171 Image Hash specified in the Activate Firmware Command (see Use Case CS06 in Section
3172 11.5 for details of the Activate Firmware Command Payload construction).

3173 If the two Hashes match, the Device shall attempt to activate the firmware image.

3174 If the two Hashes do not match, the Device shall not attempt to activate the firmware image.

3175 The Device shall issue a relevant Activate Firmware Response detailing the success or
3176 failure (see Use Case CS06 in Section 11.5 for details of the Activate Firmware Command
3177 Payload construction).

3178 **11.3 CS05a Distribute Firmware to Communications Hub**

3179 This Use Case covers the distribution of an Upgrade Image that is intended for a
3180 Communications Hub to that Communications Hub.

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Command and Response
Message Type Category	None – this is a Variant Message
Capable of future dated invocation?	No
Protection Against Replay Required?	No
SEC User Interface Services Schedule (Service Request) Reference	N/A (this Command is not available as part of the User Interface Services Schedule)
Valid Target Device(s)	Communications Hub
Valid Business Originator role(s) for Command invocation (and so, for DLMS COSEM Commands, which Application Association 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

3181 Table 11.3: Use Case Cross References for CS05a Distribute Firmware to Communications Hub

3182 11.3.1 Pre-conditions

3183 None.

3184 11.3.2 Detailed Steps

3185 The Upgrade Image shall be constructed according to Section 11.2.2.

3186 The Upgrade Image shall be transported to the Communications Hub.

3187 The Communications Hub shall verify the Upgrade Image according to Section 11.2.5, verify
 3188 the Upgrade Image is suitable for this Communications Hub, and update its Event Log with
 3189 the outcome of that verification.

3190 11.4 CS05b Distribute Firmware to ESME / GSME

3191 This Use Case covers the distribution of an OTA Upgrade Image that is intended for a
 3192 GSME or ESME to that GSME or ESME.

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Command and Response
Message Type Category	None – this is a Variant Message
Capable of future dated invocation?	No
Protection Against Replay Required?	No
SEC User Interface Services Schedule (Service Request) Reference	11.1
Valid Target Device(s)	ESME / GSME

Valid Business Originator role(s) for Command invocation (and so, for DLMS COSEM Commands, which Application Association 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

3193 Table 11.4: Use Case Cross References for CS05b Distribute Firmware to ESME / GSME

3194 11.4.1 Pre-conditions

3195 None.

3196 11.4.2 Detailed Steps

3197 Italicised terms in this Section 11.4.2 shall have the meaning specified in ZSE.

3198 ESME and GSME shall use the ZCL *Image Block Request* and *Image Block Response*
3199 commands to retrieve available OTA images.

3200 ESME and GSME shall not use the ZCL *Query Specific File Request* and *Query Specific File*
3201 *Response* commands.

3202 The OTA Upgrade Image shall be populated according to Section 11.2.3.

3203 The OTA Upgrade Image shall be transported to the Communications Hub through which
3204 the Device communicates.

3205 The Communications Hub shall update its OTA Cluster to reflect availability of the OTA
3206 Upgrade Image, once the image is received by the Communications Hub.

3207 The Communications Hub, as OTA Server, shall indicate availability of an OTA Upgrade
3208 Image differently for ESME and GSME:

- 3209 • for ESME, the Communications Hub shall send a ZSE *Image Notify* command; and
- 3210 • for GSME, the Communications Hub shall set a *the New OTA Firmware flag* (Bit
3211 Number 0) in *FunctionalNotificationFlags*.

3212 The ESME / GSME shall download an OTA Upgrade Image when it is aware of the
3213 availability of a suitable OTA Upgrade Image using the *QueryNextImage* and *Image*
3214 *Block/Page* commands specified in the *OTA Cluster* specification²².

3215 The ESME / GSME shall verify the Upgrade Image contained within the OTA Upgrade
3216 Image according to Section 11.2.5, and update its Event Log with the outcome of that
3217 verification.

3218 If the verification is successful, the ESME / GSME shall construct and send a Firmware
3219 Distribution Receipt Alert, according to Section 11.2.6, and shall store the Manufacturer
3220 Image contained within the OTA Upgrade Image.

3221 If the verification is not successful, the Device shall discard the OTA Upgrade Image, and
3222 send a Firmware Distribution Receipt Alert, as detailed in Section 11.2.6.

²² ZigBee Document 09-5264r23

3223 On receipt of a Firmware Distribution Receipt Alert, the Supplier may verify the cryptographic
3224 protection as specified in Section 6.8.3.

3225 Additionally, the Supplier may verify that the Manufacturer Image received by the Device is
3226 that intended by comparing the Manufacturer Image Hash in the Firmware Distribution
3227 Receipt Alert, with the Hash which it calculates over the Manufacturer Image provided.

3228 11.5 CS06 Activate Firmware

3229 This Use Case covers the activation of a Firmware Image.

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Command, Response and Alert (if future dated)
Message Type Category	SME.C.C
Capable of future dated invocation?	Yes
Protection Against Replay Required?	No
SEC User Interface Services Schedule (Service Request) Reference	N/A for Communications Hub (this Command is not available as part of the User Interface Services) 11.3 for ESME and GSME
Valid Target Device(s)	ESME / GSME / CH
Valid Business Originator role(s) for Command invocation (and so, for DLMS COSEM Commands, which Application Association is to be used for delivery of the Command to the Device) [Remote Party Messages Only]	Supplier for ESME / GSME WAN Provider for CH
Valid Response Recipient role(s) (only for Messages authorised by the Access Control Broker on behalf of parties not known to the Device) [Remote Party Messages Only]	N/A
Valid initiating Device type(s) [HAN Only Messages]	N/A
Protocol	ASN.1

3230 Table 11.5: Use Case Cross References for CS06 Activate Firmware

3231 11.5.1 Pre-conditions

3232 None.

3233 11.5.2 Detailed Steps

3234 11.5.2.1 Construction of Command

3235 Activate Firmware Command Payloads shall be constructed according to the requirements
3236 of Section 11.5.2.3 and populated as specified in Table 11.5.2.1.

3237 MAC Header, Grouping Header, KRP Signature and ACB-SMD MAC shall be populated as
3238 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	Generalized Time	The date-time at which the Command is to be executed, if future dated	OPTIONAL	

3239 Table 11.5.2.1: @ActivateFirmware.CommandPayload population

3240 **11.5.2.2 Device processing of Command and Response handling**

3241 The Device receiving an Activate Firmware Command shall undertake processing steps in
3242 the sequence defined in this Section 11.5.2.2.

3243 The Device shall:

- 3244 1. undertake Command Authenticity and Integrity Verification as required for a Command
3245 of the SME.C.C Message Category;
- 3246 2. if `executionDateTime` is present then the Device shall:
 - 3247 ○ record `manufacturerImageHash`, `originatorCounter` and
3248 `executionDateTime`;
 - 3249 ○ construct and send a Response where `executionOutcome` is not present.
3250 Grouping Header is constructed and Response Cryptographic Protection is
3251 applied as required for a Response of the SME.C.C Message Categories; and
 - 3252 ○ at the date-time specified in `executionDateTime`, undertake the processing
3253 from step 3.
- 3254 If `executionDateTime` is not present then the Device shall continue processing from
3255 step 3 immediately;
- 3256 3. if the Device does not have a stored Manufacturer Image then set
3257 `activateImageResponseCode` to `noImageHeld` and process from step 7;
- 3258 4. calculate Manufacturer Image Hash. If the calculated value does not equal
3259 `manufacturerImageHash` then the Device shall set `activateImageResponseCode`
3260 to `hashMismatch` and process from step 7;
- 3261 5. attempt to activate Manufacturer Image. If the activate fails then the Device shall set
3262 `activateImageResponseCode` to `activationFailure` and process from step 7;
- 3263 6. set `activateImageResponseCode` to `success`;
- 3264 7. populate the `executionOutcome` according to the requirements of Section 11.5.2.3
3265 using the `activateImageResponseCode` value produced by the processing in this

- 3266 Section 11.5.2.2, the value of `originatorCounter` from the Command and the
 3267 version of firmware now in operation to populate `firmwareVersion`;
- 3268 8. construct Grouping Header and apply the Response Cryptographic Protection required
 3269 for a Response / Alert of the SME.C.C / SME.A.C Message Categories respectively. In
 3270 such an Alert, the Message Code shall be 0x00CA. The Response / Alert shall be
 3271 addressed to the Business Originator of the Corresponding Command. If
 3272 `activateImageResponseCode` is success then `alertCode` shall be 0x8F66 else
 3273 `alertCode` shall be 0x8F67; and
- 3274 9. send the Response if `executionDateTime` was not present in the Command or send
 3275 the Alert if `executionDateTime` was present in the Command.

3276 On receipt of the Response, the recipient may undertake the 'Response Recipient
 3277 Verification' for Responses of type SME.C.C. or for Alerts of type SME.A.C, dependent upon
 3278 the Message received.

3279 *11.5.2.3 Activate Firmware Command, Response and Alert Payloads – structure* 3280 *definition*

3281 Each instance of `@ActivateFirmware.CommandPayload` and of
 3282 `@ActivateFirmware.ResponsePayload` and of
 3283 `@ActivateFirmware.AlertPayload` shall be an octet string containing the DER
 3284 encoding of the populated structure defined in this Section 11.5.2.3 which specifies the
 3285 structure in ASN.1 notation.

3286 `ActivateFirmware DEFINITIONS ::= BEGIN`

```

3287 CommandPayload ::=                               SEQUENCE
3288 {
3289   -- specify the hash of the Manufacturer Image to be activated
3290   manufacturerImageHash                          OCTET STRING,
3291
3292   -- the Originator Counter as in the Grouping Header of the Command
3293   originatorCounter                              INTEGER (0..
3294   18446744073709551615),
3295
3296   -- the date-time at which the Command is to execute, if future dated
3297   executionDateTime                              GeneralizedTime OPTIONAL
3298 }
3299
3300 ResponsePayload ::=                               CHOICE
3301 {
3302   -- if the Command is future dated, the Response will not have any details of
3303   -- execution (those will be in the subsequent alert)
3304   commandAccepted                               NULL,
3305
3306   -- if the Command is for immediate execution, the Response will detail the
3307   -- outcomes
3308   executionOutcome                              ExecutionOutcome
3309 }
3310
3311 AlertPayload ::=                               SEQUENCE
3312 {
3313   -- specify the Alert Code
3314   alertCode                                      INTEGER(0..4294967295),
3315
3316   -- specify the date-time of execution
3317   executionDateTime                              GeneralizedTime,
3318
3319   -- the Originator Counter as in the Grouping Header of the corresponding Command
3320   originatorCounter                              INTEGER (0..
3321   18446744073709551615),
3322
3323   -- detail what happened when the future dated command was executed
3324 
```

```

3325         executionOutcome                               ExecutionOutcome
3326     }
3327
3328 ExecutionOutcome ::=                               SEQUENCE
3329 {
3330     -- Specify whether the activation was successful or not
3331     activateImageResponseCode                          ActivateImageResponseCode,
3332
3333     -- Specify the Device's now current firmware version. The value shall be four
3334     octets in length and shall correspond to the File Version field in the ZSE OTA
3335     Header structure.
3336     firmwareVersion                                    OCTET STRING
3337 }
3338
3339 ActivateImageResponseCode ::= INTEGER
3340 {
3341     success                                             (0),
3342     noImageHeld                                         (1),
3343     hashMismatch                                       (2),
3344     activationFailure                                   (3)
3345 }
3346
3347 END

```

12 Requirements for Certificates

This Section 12 lays out requirements as to structure and content to which all valid authorised Certificates shall comply, in so far as those requirements affect the processing carried out by Devices. All terms in this section shall, where not defined in the GBCS, have the meanings in IETF RFC 5759²³ and IETF RFC 5280.

12.1 Requirements applicable to all Certificates

All Security Credential Documents that are successfully authorised within the APKI for use by Devices 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 5480²⁴;
- only provide for signature methods that are explicitly allowed within the GBCS. This means using P-256 Private Keys with SHA 256 and ECDSA;
- contain a `serialNumber` of no more than 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

²³ <http://tools.ietf.org/html/rfc5759>

²⁴ <http://tools.ietf.org/html/rfc5480>

- 3387 • have a valid `notBefore` field consisting of the time of issue encoded and a valid
 3388 `notAfter` as per IETF RFC 5280 section 4.1.2.5.

3389 12.2 Requirements applicable to Organisation Certificates 3390 only

3391 All Organisation Certificates that are Authorised for use by Devices shall:

- 3392 • have a fixed expiration date in the `notAfter` field which shall not be
 3393 `GeneralizedTime` value of 99991231235959Z;
- 3394 • contain a non-empty subject field which shall contain a unique X.500 Distinguished
 3395 Name (DN), which shall be the unique trading name of the Organisation, and an
 3396 `X520OrganizationalUnitName` whose value shall be set to the
 3397 `RemotePartyRole` that this Certificate allows the subject of the Certificate to
 3398 perform; and
- 3399 • contain a single Public Key except where the `RemotePartyRole` = `root`. Where
 3400 the `RemotePartyRole` = `root`, the Certificate shall contain two public keys. The
 3401 second public key shall be referred to as the Contingency Key²⁵ and shall be present
 3402 in the `WrappedApexContingencyKey` extension with the meaning of IETF RFC
 3403 5934²⁶. The Contingency Key shall be Encrypted as per the requirements of Section
 3404 13.3.5.8.1.

3405 12.3 Requirements applicable to Certificates where 3406 `RemotePartyRole` = `root` or `issuingAuthority`

3407 All Remote Parties' Certificates that:

- 3408 • are Authorised within the APKI for use by Devices; and
- 3409 • have a `X520OrganizationalUnitName` whose value is either `root` or
 3410 `issuingAuthority`
- 3411 shall:
- 3412 • have a `keyUsage` with a value of `keyCertSign` and `cRLSign`. For clarity, Devices
 3413 are not required to use the associated Public Keys in relation to the `cRLSign`
 3414 `keyUsage`;
- 3415 • where `X520OrganizationalUnitName` = `issuingAuthority`:
 - 3416 ○ contain at least one `CertPolicyId` in the `certificatePolicies` extension
 3417 that refers to the OID(s) valid for usage in GB Smart Metering;
 - 3418 ○ contain the `basicConstraints` extension, with values `cA` = `True`, and
 3419 `pathLen` = 0. This extension shall be marked as critical;
- 3420 • where `X520OrganizationalUnitName` = `root`:
 - 3421 ○ contain a single `CertPolicyId` in the `certificatePolicies` extension that
 3422 refers to the OID for `anyPolicy`; and

²⁵ The Contingency Key is a second public key held in the Root Certificate (and protected with an encryption key). Its sole purpose is to allow the validation of a specific command that allows direct replacement of the Root Trust Anchor. The command (an Apex Trust Anchor Update message) is signed with a private key (used once only, and only to sign this message) that only the second public key (known as the Contingency Key) can verify and therefore authorise action of.

²⁶ Housley, R., Ashmore, S., and C. Wallace, 'Trust Anchor Management Protocol (TAMP)', RFC 5934, August 2010.
<https://tools.ietf.org/html/rfc5934>

- 3423 ○ contain the `basicConstraints` extension, with the value `cA = True` and
- 3424 `pathLen` absent (unlimited). This extension shall be marked as critical.

12.4 Requirements applicable to Certificates where RemotePartyRole is neither root nor issuingAuthority

3428 All Remote Parties' Certificates that:

- 3429 • are Authorised within the APKI for use by Devices; and
 - 3430 • have a `X520OrganizationalUnitName` whose value is not `root` and is not
 - 3431 `issuingAuthority`
- 3432 shall:
- 3433 • contain a `subjectUniqueID` whose value shall be the 8 octet Entity Identifier of the
 - 3434 subject of the Certificate;
 - 3435 • have a `keyUsage` with a value of only one of `digitalSignature` or
 - 3436 `keyAgreement`; and
 - 3437 • contain a single `policyIdentifier` in the `certificatePolicies` extension that
 - 3438 refers to the OID applicable to the environment the Certificate has been issued in.

12.5 Requirements applicable to Device Certificates

3440 All Device Certificates that are Authorised within the APKI for use by Devices shall:

- 3441 • not have a well-defined expiration date and so the `notAfter` field shall be assigned
- 3442 the `GeneralizedTime` value of `99991231235959Z`;
- 3443 • have an empty `SubjectName`;
- 3444 • have a `keyUsage` with a value of only one of `digitalSignature` or
- 3445 `keyAgreement`;
- 3446 • contain a single `policyIdentifier` in the `certificatePolicies` extension that
- 3447 refers to the OID applicable to the environment the Device Certificate has been
- 3448 issued in;
- 3449 • contain a `SubjectAltName` extension which shall contain a single `GeneralName` of
- 3450 type `OtherName` that is further sub-typed as a `HardwareModuleName` (`id-on-`
- 3451 `HardwareModuleName`) as defined in IETF RFC 4108²⁷. The `hwSerialNum` field
- 3452 shall be set to the Device's Entity Identifier. In adherence to IETF RFC 5280,
- 3453 `SubjectAltName` shall be marked as critical; and
- 3454 • contain a single Public Key.

12.6 Device processing of Certificates

3456 In relation to Certificates, Devices shall:

- 3457 • accept unexpected (not required by the GBCS) certificate extensions and shall ignore
- 3458 silently non-critical unrecognized certificate extensions;
- 3459 • in adherence with the requirements of IETF RFC 5280, reject any certificate
- 3460 containing unrecognized critical certificate extensions; and

²⁷ <http://tools.ietf.org/html/rfc4108>

- 3461 • reject any certificate containing either policy mappings or name constraints.

13 Managing Security Credentials on Devices

13.1 Introduction – informative

This Section 13 includes the Use Cases related to the management of Security Credentials on Devices in terms of the relevant Commands, Responses and Alerts:

- 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); and
- Section 13.8 - GPF Device Log Backup and Restore (GCS59 and GCS62).

13.1.1 Device Security Credentials – informative

In terms of processing relating to a Device's own Security Credentials:

- the Command to Devices for issuing Device 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:

- 3502 • the Certificate's expiry (Organisation Certificates will only be valid for a fixed period of
3503 time);
- 3504 • the Known Party transferring control to a different organisation (for example on
3505 Change of Supplier);
- 3506 • the cryptographic algorithms, or parameters such as key length, needing to be
3507 changed;
- 3508 • the Known Party having lost the use of the corresponding Private Key; or
- 3509 • there being concerns that someone other than the Known Party has use of, or may
3510 have use of, the corresponding Private Key.

3511 Thus, an 'Update Security Credentials Command' must be supported by all Devices that rely
3512 on Remote Party Security Credentials to act as Trust Anchors. Related, all such Devices
3513 need to support a 'Provide Security Credential Details' Command, so that Remote Parties
3514 can be sure which Devices need to have credentials replaced.

3515 However, if these Trust Anchors could be replaced without proper protections, attackers
3516 could take over control of Devices or the Devices could be rendered inoperable. Thus, a
3517 Device needs to do thorough checks before applying an Update Security Credentials
3518 Command. The checks that the Device can and must do vary dependent on the reasons for
3519 the change. Thus, Section 13.2.1 lays out a number of different checks and the
3520 circumstances in which corresponding Commands may be issued. Broadly the following
3521 checks are carried out by the Device:

- 3522 • is the Command properly formed?
- 3523 • is the Command for the Device that it has been delivered to, and is the Command
3524 one that it has not processed previously?
- 3525 • are the Remote Parties apparently authorising the Command allowed to authorise it?
- 3526 • was the Command Authorised by the Remote Parties that it appears to be Authorised
3527 by?
- 3528 • were the Certificates in the payload of the Command issued by properly Authorised
3529 parties, specifically by Certification Authorities Authorised (by 'root' under the APKI)
3530 to issue GB Smart Metering Certificates?

3531 Only when a Device has successfully undertaken all five sets of checks should it action the
3532 Update Security Credentials Command.

3533 Other Critical Commands only have to complete the first four categories of check.

3534 **13.1.3 Trust Anchor Management Protocol (TAMP) – informative**

3535 The GBCS does not specify a fully compliant TAMP solution due to the limited processing
3536 and networking capability of Devices. However, it does incorporate checks that are
3537 functionally derived from relevant checks in IETF RFC 5934.

3538 The GBCS only permits a restricted subset of 'IETF RFC 5934 like' functionality:

- 3539 • replacement of Trust Anchors is required (and specified in this Use Case) but their
3540 addition, change or removal is not allowed;
- 3541 • status queries are supported (and are specified in this Section 13.1.3); and
- 3542 • 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) 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
SEC User Interface Services Schedule (Service Request) Reference	6.24
Valid Target Device(s)	ESME / GSME / GPF / CHF / HCALCS / PPMID
Valid Business Originator role(s) for Command invocation (and so, for DLMS COSEM Commands, which Application Association is to be used for delivery of the Command to the Device) [Remote Party Messages Only]	Supplier Network Operator Access Control Broker Transitional Change of Supplier WAN Provider Recovery
Valid Response Recipient role(s) (only for Messages Authorised by the Access Control Broker on behalf of parties not known to the Device) [Remote Party Messages Only]	N/A
Valid initiating Device type(s) [HAN Only Messages]	N/A
Protocol	ASN.1

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 Protection Against Replay counter details held on the Device.

3558 **13.2.3.2 The ‘Provide Security Credential Details’ Command and Response**

3559 This Section 13.2.3.2 summarises the structure of the Provide Security Credential Details Command.

3560 If protected by an Access Control Broker MAC as per Section 13.2.4.2, a Provide Security Credential Details Command shall be the
3561 concatenation:

3562 MAC Header || Grouping Header || @ProvideSecurityCredentialDetails.Command || 0x00 || ACB-SMD MAC

3563 If protected by a KRP Signature as per Section 13.2.4.2, a Provide Security Credential Details Command shall be the concatenation:

3564 Grouping Header || @ProvideSecurityCredentialDetails.Command || 0x40 || KRP Signature

3565 If an SMD Signature is required as per Section 13.2.4.5, a Provide Security Credential Details Response shall be the concatenation:

3566 Grouping Header || @ProvideSecurityCredentialDetails.Response || 0x40 || SMD Signature

3567 If an SMD Signature is not required as per Section 13.2.4.5, a Provide Security Credential Details Response shall be the concatenation:

3568 MAC Header || Grouping Header || @ProvideSecurityCredentialDetails.Response || 0x00 || SMD-KRP MAC

3569 Where:

- 3570 • @ProvideSecurityCredentialDetails.Command and Response shall each be an octet string containing the DER encoding of
3571 the populated ASN.1 structure (as laid out in Section 13.2.3.3);
- 3572 • 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
3573 SMD or KRP Signature is not present;
- 3574 • KRP Signature and ACB-SMD MAC are as defined in Section 13.2.4.2;
- 3575 • SMD Signature and SMD-KRP MAC are as defined in Section 13.2.4.5; and
- 3576 • MAC Header and Grouping Header are as defined in Section 7.2.

3577 **13.2.3.3 The @ProvideSecurityCredentialDetails.Command and @ProvideSecurityCredentialDetails.Response structure definition**

3578 Each instance of @ProvideSecurityCredentialDetails.Command and of @ProvideSecurityCredentialDetails.Response shall
3579 be an octet string containing the DER²⁸ encoding of the populated structure defined in this Section 13.2.3.3 which specifies the structure in
3580 ASN.1 notation²⁹.

3581 ProvideSecurityCredentialDetails DEFINITIONS ::= BEGIN

²⁸ <https://www.itu.int/rec/T-REC-X.690/en>

²⁹ <https://www.itu.int/rec/T-REC-X.680/en>

```

3582
3583 Command ::=
3584 {
3585     -- Identify which of the Public Keys on the Device is to be used in verifying the Signature or MAC
3586     -- (so defining the nature of the verification by way of the KeyUsage parameter held on the
3587     -- Device for the Public Key so identified).
3588
3589     authorisingRemotePartyTACellIdentifier      TrustAnchorCellIdentifier,
3590
3591     -- List the Remote Party Role(s) for which credential details are required
3592
3593     remotePartyRolesCredentialsRequired        SEQUENCE OF RemotePartyRole
3594 }
3595
3596 Response ::=
3597     SEQUENCE OF RemotePartyDetails
3598
3599 RemotePartyDetails ::=
3600     SEQUENCE
3601 {
3602     -- Which Remote Party do these details relate to?
3603     remotePartyRole                          RemotePartyRole,
3604
3605     -- statusCode shall be success unless the role is not valid on this type of Device or there is a processing failure
3606     statusCode                              StatusCode,
3607
3608     -- What is the current Update Security Credentials Protection Against Replay number on the Device for this role, where there is
3609     -- such a number for this role?
3610
3611     currentSeqNumber                        SeqNumber OPTIONAL,
3612
3613     -- What are the details held on the Device for each of the Cells related to this role? The list shall have between one and
3614     -- three entries (e.g. there will be one if role is transitional change of supplier; there may be three if role is supplier)
3615
3616     trustAnchorCellsDetails                SEQUENCE OF TrustAnchorCellContents OPTIONAL
3617 }
3618
3619 SeqNumber ::=
3620     INTEGER (0.. 18446744073709551615)
3621
3622 TrustAnchorCellContents ::=
3623     SEQUENCE
3624 {
3625     -- To what cryptographic use can the Public Key in this Cell be put? Some Remote Party Roles
3626     -- (e.g. supplier) can have more than one Public Key on a Device and each one would only have
3627     -- a single cryptographic use.

```



```

3626
3627 trustAnchorCellKeyUsage          KeyUsage,
3628
3629 -- trustAnchorCellUsage is to allow for multiple Public Keys of the same keyUsage for the same Remote
3630 -- Party Role. This will be absent except where used to refer to the Supplier Key Agreement Key.
3631 -- This Key is used solely in relation to validating Supplier generated MACs on Prepayment Top Up transactions.
3632
3633 trustAnchorCellUsage              CellUsage DEFAULT management,
3634
3635 -- The subjectUniqueID which shall be the 64 bit Entity Identifier of the Security Credentials in this Trust Anchor Cell.
3636
3637 existingSubjectUniqueID           OCTET STRING,
3638
3639 -- The APKI requirements mean that KeyIdentifier attributes will all be 8 byte SHA-1 Hashes.
3640 -- existingSubjectKeyIdentifier shall be set accordingly based on the contents of the Trust Anchor Cell
3641
3642 existingSubjectKeyIdentifier       OCTET STRING
3643 }
3644
3645 TrustAnchorCellIdentifier ::=      SEQUENCE
3646 {
3647 -- Which Remote Party Role does this Cell relate to?
3648
3649 trustAnchorCellRemotePartyRole    RemotePartyRole,
3650
3651 -- To what cryptographic use can the Public Key in this Cell be put? Some Remote Party Roles
3652 -- (e.g. supplier) can have more than one Public Key on a Device and each one would only have
3653 -- a single cryptographic use.
3654
3655 trustAnchorCellKeyUsage            KeyUsage,
3656
3657 -- trustAnchorCellUsage is to allow for multiple Public Keys of the same keyUsage for the same Remote
3658 -- Party Role. This may be absent except where use to refer to the Supplier Key
3659 -- Agreement Key used solely in relation to validating Supplier generated MACs on Prepayment Top Up transactions
3660
3661 trustAnchorCellUsage              CellUsage DEFAULT management
3662 }
3663
3664 CellUsage ::=                     INTEGER {management(0), prePaymentTopUp(1)}
3665
3666 RemotePartyRole ::=               INTEGER
3667 {
3668 -- Define the full set of Remote Party Roles in relation to which a Device may need to undertake
3669 -- processing. Note that most Devices will only support processing in relation to a subset of these.

```

```

3670
3671 root (0),
3672 recovery (1),
3673 supplier (2),
3674 networkOperator (3),
3675 accessControlBroker (4),
3676 transitionalCoS (5),
3677 wanProvider (6),
3678 issuingAuthority (7), -- Devices will receive such Certificates but they do not
3679 -- need to store them over an extended period
3680
3681
3682
3683 -- The 'other' RemotePartyRole is for a party whose role does not allow it to invoke any Device function apart from
3684 -- UpdateSecurityCredentials. This is to allow for Device functionality to be locked out of usage until a valid
3685 -- Remote Party can be identified e.g. where roles cannot be fixed until a Device is bought in to operation
3686 other (127)
3687
3688 }
3689
3690 -- KeyUsage is only repeated here for ease of reference. It is defined in RFC 5912
3691
3692 KeyUsage ::= BIT STRING
3693 {
3694 -- Define valid uses of Public Keys.
3695
3696 digitalSignature (0),
3697 contentCommitment (1), -- not valid for GBCS compliant transactions
3698 keyEncipherment (2), -- not valid for GBCS compliant transactions
3699 dataEncipherment (3),
3700 keyAgreement (4),
3701 keyCertSign (5),
3702 cRLSign (6),
3703 encipherOnly (7),
3704 decipherOnly (8) -- not valid for GBCS compliant transactions
3705 }
3706
3707 -- The GBCS only allows for a constrained set of Trust Anchor Cell operations and so the list of possible outcomes
3708 -- is more limited than in IETF RFC 5934. The list below is that more constrained subset
3709
3710 StatusCode ::= ENUMERATED {
3711
3712 success (0),
3713

```

```

3714 -- trustAnchorNotFound indicates that details of a trust anchor were requested, but the referenced trust anchor
3715 -- is not represented on the Device
3716
3717 trustAnchorNotFound          (25),
3718
3719 other                        (127)}
3720
3721
3722 END

```

3723 13.2.4 Provide Security Credential Details from a Device – Processing Steps

3724 This Section 13.2.4 lays out the requirements relating to the construction, protection and Authentication of the Provide Security Credentials
 3725 Command, and the construction, protection and Authentication of the corresponding Response.

3726 13.2.4.1 Command Construction

3727 The Remote Party constructing the Command shall populate Grouping Header according to the requirements of Section 7.2.6.

3728 @ProvideSecurityCredentialDetails.Command shall have the structure defined in Section 13.2.3.3, and the Remote Party constructing
 3729 the Command shall populate with values according to Table 13.2.4.1.

3730 The Remote Party constructing the Command shall populate Command Length once it has fully populated
 3731 @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) ,	Mandatory if authorising RemoteParty	The role of the Party applying the Command's cryptographic protection

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
		accessControlBroker (4) , transitionalCoS (5) , wanProvider (6)	TACellIdentifier present	
trustAnchorCellKeyUsage	BIT STRING	digitalSignature (0) , keyAgreement (4)	Mandatory if authorising RemoteParty TACellIdentifier present	Where the Command's cryptographic protection is a digital signature (digitalSignature) or a MAC (keyAgreement). The value shall be digitalSignature unless trustAnchorCellRemotePartyRole = accessControlBroker
trustAnchorCellUsage	INTEGER	management (0)	DEFAULT management	Must be absent or set to 'management' since the prePaymentTopUp key pair cannot be used in relation to this command
remotePartyRolesCredentialsRequired	SEQUENCE OF			
RemotePartyRole	INTEGER	root (0) , recovery (1) , supplier (2) , networkOperator (3) , accessControlBroker (4) , transitionalCoS (5) , wanProvider (6)	Mandatory to have at least one	List the Remote Party Role(s) for which credential details are required

3732 Table 13.2.4.1: Attribute values for Provide Security Credentials Command

3733 **13.2.4.2 Command Cryptographic Protection**

3734 If the Access Control Broker is undertaking this step to apply a MAC, then the Access Control Broker shall undertake the steps in Section
3735 13.2.4.2.1 otherwise:

- 3736 • the Remote Party originating the Command shall generate a Signature for the Command and set KRP Signature accordingly;
- 3737 • the Signature, for incorporation in the Command, shall only be generated once all fields of the Grouping Header ||
- 3738 @ProvideSecurityCredentialDetails.Command are populated as per the requirements for the Command Construction stage;
- 3739 and
- 3740 • the Remote Party shall use its Private Digital Signing Key to generate the Signature.

3741 13.2.4.2.1 Access Control Broker MAC

3742 If the Access Control Broker is undertaking this step to apply a MAC, then the Access Control Broker shall calculate a MAC using the
 3743 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	

3744 Table 13.2.4.2.1: Calculation of Access Control Broker MAC for Provide Security Credentials command

3745 13.2.4.3 Command Authenticity and Integrity Verification

3746 The Device shall undertake processing according to the requirements of this section before undertaking any other processing of the Command.

3747 The checks should be carried out in the order specified. The Device shall cease checking at the point that any one check fails.

3748 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 <code>authorisingRemotePartyTACellIdentifier</code>	As specified in Section 13.2.4.3.1

3749 Table 13.2.4.3: Provide Security Credentials Command authenticity and integrity verification

3750 Should any of the checks detailed in this Section 13.2.4.3 fail then the Device shall:

- 3751 • generate an entry in the Security Log recording failed Authentication;
- 3752 • discard the Command without execution and without sending a Response; and
- 3753 • send an Alert notifying the failed Authentication, constructed as specified in Section 6.2.4.2, populated with the relevant Alert Code from
- 3754 Section 16 , to the Known Remote Party identified by the Security Credentials it holds in the `{supplier, management,`
- 3755 `digitalSignature}` Trust Anchor Cell.

3756 Where all of the checks detailed in this Section 13.2.4.3 succeed the Device shall process the Command and produce a Response.

3757 *13.2.4.3.1 Command Authenticity and Integrity Verification*

3758 The Device shall undertake the following checks until either all are successful or one has failed.

- 3759 1. If `trustAnchorCellUsage` is present it has a value of management else this test shall fail.
- 3760 2. If `trustAnchorCellKeyUsage = keyAgreement` then
- 3761 ((`trustAnchorCellRemotePartyRole = accessControlBroker`) and (the MAC calculated by the Device according to Table
- 3762 13.2.4.3.1 equates to ACB-SMD MAC)
- 3763 else
- 3764 ((`trustAnchorCellKeyUsage = digitalSignature`) and (the Device shall use the Public Key in the Trust Anchor Cell
- 3765 identified by `authorisingRemotePartyTACellIdentifier` to verify that KRP Signature is the Digital Signature across
- 3766 Grouping Header || `@ProvideSecurityCredentialDetails.Command`)
- 3767 else
- 3768 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	

3769 Table 13.2.4.3.1: Calculation of MAC for Provide Security Credential Details Command

3770 **13.2.4.4 Response Construction**

3771 The Device shall populate Grouping Header according to the requirements of Section 7.2.6.

3772 The @ProvideSecurityCredentialDetails.Response shall have the structure defined in Section 13.2.3.3, and the Device shall
3773 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) ,	Mandatory if SEQUENCE is present	The role to which the credentials in this SEQUENCE relate

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
		transitionalCoS (5) , wanProvider (6) ,		
statusCode	ENUMERATED	success (0) , trustAnchorNotFound (25) , other (127)	Mandatory if SEQUENCE is present	Whether the Device can supply the details
currentSeqNumber	INTEGER	The corresponding Counter value	Present if statusCode=0	The Protection Against Replay number held by the Device for this role's use of the Update Security Credentials Command
trustAnchorCellsDetails	SEQUENCE OF		At least one in the SEQUENCE OF must be present if statusCode=0	
SEQUENCE				
trustAnchorCellKeyUsage	BIT STRING	digitalSignature (0) , keyAgreement (4) , keyCertSign (5)	Mandatory if SEQUENCE is present	To what use can the public key in this Cell be put
trustAnchorCellUsage	INTEGER	prePaymentTopUp(1)	DEFAULT management (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

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
existingSubjectKeyIdentifier	OCTET STRING	Key Identifier of the key in this Cell	Mandatory if SEQUENCE is present	

3774 Table 13.2.4.4: Attribute values for Provide Security Credentials Response

3775 **13.2.4.5 Response Cryptographic Protection**

3776 If the Command that triggered this Response was protected by a MAC then the Device shall calculate a MAC using the parameters in Table
3777 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	

3778 Table 13.2.4.5: Calculation of MAC for Provide Security Credential Details Response

3779 Otherwise:

- 3780 • the Device creating the Response shall generate a Signature for the Response and set SMD Signature to the value calculated;
- 3781 • the Signature, for incorporation in the Response, shall only be generated once all fields of the Grouping Header || Length ||
- 3782 @ProvideSecurityCredentialDetails.Response are populated, as per requirements for the Response Construction stage; and
- 3783 • the Device shall use its Private Digital Signing Key to generate the Signature.

3784 **13.2.4.6 Response Recipient Cryptographic Verification**

3785 If the Response contains a MAC, the Access Control Broker can verify that MAC by calculating a MAC according to the parameters in Table
 3786 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	

3787 Table 13.2.4.6: Calculation of MAC for Provide Security Credential Details Response Verification

3788 **13.3 CS02b Update Security Credentials Command, Response and Alert**3789 **13.3.1 Description**

3790 This Section 13.3 covers the creation, validation and processing of:

- 3791 • Update Security Credentials Commands;
- 3792 • Responses to such Commands; and
- 3793 • Alerts resulting from the future dated execution of such Commands.

3794 The Update Security Credentials Command shall be:

- 3795 • used solely to replace Remote Party Security Credentials held in Trust Anchor Cells on Devices;
- 3796 • supported by any Device that can process Remote Party Messages; and
- 3797 • the only Command that Devices are capable of accepting for replacement of Remote Party Security Credentials, once the tamper seal is
 3798 applied to the Device.

3799 **13.3.2 Use Case Cross References**

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Command and Response / Alert
Message Type Category	Variant Message and is Critical
Capable of future dated invocation?	Yes (but see constraint in Table 13.3.5.1, check 1.3)
Protection Against Replay Required?	The Protection Against Replay mechanisms for Update Security Credentials are specified in Section 13.3. The Protection Against Replay mechanisms of other sections of the GBCS do not apply
SEC User Interface Services Schedule (Service Request) Reference	6.15, 6.23, 8.5, 6.21
Valid Target Device(s)	ESME / GSME / GPF / CHF / HCALCS / PPMID
Valid Business Originator role(s) for Command invocation (and so, for DLMS COSEM Commands, which Application Association is to be used for delivery of the Command to the Device) [Remote Party Messages Only]	Supplier Network Operator Access Control Broker Transitional Change of Supplier WAN Provider Recovery
Valid Response Recipient role(s) (only for Messages Authorised by the Access Control Broker on behalf of parties not known to the Device) [Remote Party Messages Only]	N/A
Valid initiating Device type(s) [HAN Only Messages]	N/A
Protocol	ASN.1

3800 Table 13.3.2: Use Case Cross References for Update Security Credentials Command

3801 13.3.3 Command, Response and Alert Structure

3802 13.3.3.1 The Update Security Credentials Command

3803 This Section 13.3.3.1 summarises the structure of the Update Security Credentials Command, which depends on
3804 `credentialsReplacementMode` and the `deviceType` of the Device.

3805 If `credentialsReplacementMode` = `anyByContingency` or `anyExceptAbnormalRootByRecovery` then an Update Security Credential
3806 Details Command shall be the concatenation:

3807 Grouping Header || `@UpdateSecurityCredentials.CommandPayload` || 0x40 || KRP Signature

3808 If `credentialsReplacementMode` = `accessControlBrokerByACB` and `deviceType` is not
3809 `communicationsHubCommunicationsHubFunction` then an Update Security Credentials Command shall be the concatenation:

3810 MAC Header || Grouping Header || `@UpdateSecurityCredentials.CommandPayload` || 0x00 || ACB-SMD MAC

3811 In all other cases, the Update Security Credentials Command shall either be the concatenation:

3812 MAC Header || Grouping Header || `@UpdateSecurityCredentials.CommandPayload` || 0x40 || KRP Signature|| ACB-SMD MAC

3813 In these Command structures:

- 3814 • `@UpdateSecurityCredentials.CommandPayload` shall be an octet string containing the DER encoding of the populated ASN.1
3815 structure (as laid out in Section 13.3.5.11);
- 3816 • Grouping Header shall be constructed as specified in Section 7.2.7 with Business Originator ID being the Entity Identifier of the Known
3817 Remote Party which generated KRP Signature, and with Business Originator Counter being that of the same Known Remote Party;
- 3818 • KRP Signature shall be generated as specified in Section 6.3.3;
- 3819 • ACB Grouping Header shall be constructed as specified in Section 7.2.7 with Business Originator ID being the Entity Identifier of the
3820 Access Control Broker and Business Originator Counter being that of the Access Control Broker;
- 3821 • MAC Header shall be constructed as specified in Section 7.2.5; and
- 3822 • ACB-SMD MAC shall be calculated as specified in Section 6.2.3.

3823 13.3.3.2 The Update Security Credentials Response

3824 An Update Security Credentials Response shall be the concatenation:

3825 Grouping Header || `@UpdateSecurityCredentials.ResponsePayload` || 0x40 || SMD Signature

3826 where:

- 3827 • `@UpdateSecurityCredentials.ResponsePayload` shall be an octet string containing the DER encoding of the populated ASN.1
3828 structure (as laid out in Section 13.3.5.11);
- 3829 • Grouping Header in the Response shall be constructed as specified in Section 7.2.7 with Business Target ID being the Entity Identifier
3830 specified in the corresponding Command's Grouping Header; and
- 3831 • SMD Signature shall be generated as specified in Section 6.3.5.

3832 *13.3.3.3 The Update Security Credentials Alert*

3833 An Update Security Credentials Alert shall be the concatenation:

3834 Grouping Header || `@UpdateSecurityCredentials.AlertPayload` || 0x40 || SMD Signature

3835 where:

- 3836 • `@UpdateSecurityCredentials.AlertPayload` shall be an octet string containing the DER encoding of the populated ASN.1
3837 structure (as laid out in Section 13.3.5.11);
- 3838 • Grouping Header in the Alert shall be constructed as specified in Section 7.2.7 with Business Target ID being the Entity Identifier
3839 specified in the corresponding Command's Grouping Header;
- 3840 • the Message Code being 0x00CB; and
- 3841 • SMD Signature shall be generated as specified in Section 6.3.5.

3842 *13.3.3.4 The Update Security Credentials Command, Response and Alert – informative*

3843 The `@UpdateSecurityCredentials.CommandPayload` structure has four parts:

- 3844 • `authorisingRemotePartyControl`: which includes details of what kind of credential replacement this Command is, which Remote
3845 Parties are authorising it and information to support Protection Against Replay protections;
- 3846 • `replacements`: which is a list of new Certificates the Device is to store details from, along with which Trust Anchor Cell each set of
3847 details is to be stored in on the Device;
- 3848 • `certificationPathCertificates`: which is a list of Certification Authority Certificates the Device will need to use in checking that
3849 the replacement Certificates were properly issued; and
- 3850 • `executionDateTime`: which, if present, specifies the date-time at which the `certificates` in the `CommandPayload` are to be used
3851 to replace the credentials currently in use on the Device. If this field is not present, the Command shall be executed immediately. If this

3852 field has the value equivalent to 'never' (which is '99991231235959Z') the certificate replacement will never happen. This is to allow
 3853 cancellation of future dated Commands. Note that future dating is not supported where certificates are being replaced in exception
 3854 conditions.

3855 The @UpdateSecurityCredentials.Response structure contains, for immediate execution commands, a list detailing the success of
 3856 failure of each of the replacements, including details of the parties affected. For future dated commands,
 3857 @UpdateSecurityCredentials.AlertPayload structure contains the list detailing the success, or failure, of each of the replacements,
 3858 including details of the parties affected.

3859 Section 13.3.5.11 contains narrative for each of the parts of these ASN.1 structures.

3860 Section 18.2.1.2 provides an illustrative instantiation of @UpdateSecurityCredentials.CommandPayload and its corresponding DER
 3861 encoding.

3862 13.3.4 Updating Security Credentials on a Device – Processing Steps

3863 This section lays out the requirements for the construction, protection and Authentication of the Update Security Credentials Command
 3864 Payload, the processing required on the Device of the Command, the construction of the corresponding Response Payload and, where
 3865 required, the Alert Payload.

3866 13.3.4.1 Command Payload construction

3867 The @UpdateSecurityCredentials.CommandPayload shall have the structure defined in Section 13.3.5.11, and the Remote Party
 3868 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,

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
				if the command changes the Remote Party in control
credentialsReplacementMode	INTEGER	rootBySupplier (0) , rootByWanProvider (1) , supplierBySupplier (2) , networkOperatorByNetworkOperator (3) , accessControlBrokerByACB (4) , wanProviderByWanProvider (5) , transCoSByTransCoS (6) , supplierByTransCoS (7) , anyExceptAbnormalRootByRecovery (8) , anyByContingency (9)	Mandatory	Specify the replacement mode so that the Device can check that the Remote Party Role authorising the command is allowed to authorise this type of replacement(s) and that all replacements in the payload are allowed within this replacement mode. The structure of the label is <i>kindOfCertificate(s)BeingReplacedBypartydoingthereplacement</i> . For example, rootBySupplier is where a new root Certificate is being provided to the Device by its Supplier
plaintextSymmetricKey	[0] IMPLICIT OCTET STRING	The symmetric key that will decrypt the encrypted Contingency Key held on the Device	OPTIONAL	Only to be present if the Contingency Key arrangements are being used (so if credentialsReplacementMode = anyByContingency). The contents provide the symmetric key to decrypt the Contingency Public Key in 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
				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. root 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) , networkOperator (3) , accessControlBroker (4) ,	Mandatory if authorisingRemotePartyTACellIdentifier present	The role of the Party applying KRP Signature. Note that where root is used, this refers only to the encrypted Contingency key in the root TA Cell, so is only valid if credentialsReplacementMode = anyByContingency and plaintextSymmetricKey is

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)		populated with the symmetric key required to decrypt that public key
trustAnchorCellKeyUsage	BIT STRING	digitalSignature (0)	Mandatory if authorisingRemotePartyTACellIdentifier present	KRP Signature is a digital signature
trustAnchorCellUsage	INTEGER	management (0)	DEFAULT management	Must be absent or set to 'management' since the prePaymentTopUp key pair cannot be used in relation to this Command
authorisingRemotePartySeqNumber	[3] IMPLICIT INTEGER	Originator Counter of Remote Party authorising the Command	Mandatory	Specify the Originator Counter for the Remote Party applying KRP Signature, or (for the Access Control Broker changing its credentials) the Access Control Broker's Originator Counter
newRemotePartyFloorSeqNumber	[4] IMPLICIT INTEGER	Originator Counter of Remote Party who will have control of this Remote Party Role if the update is successful	OPTIONAL	If the Command is to effect a change of control, then newRemotePartyFloorSeqNumber should be included and will be the value used to prevent replay of Update Security Credentials Commands, and other Commands, for the new controlling Remote Party
newRemotePartySpecialistFloorSeqNumber	[5] IMPLICIT SEQUENCE OF		OPTIONAL	Some Commands on the Device may use a different Originator Counter sequence for Protection Against Replay. The only

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
				example is the Prepayment Top Up Command on ESME and GSME. The <code>SpecialistSeqNumber</code> 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				
<code>seqNumberUsage</code>	INTEGER	<code>prepaymentTopUp</code> (0)	Mandatory if <code>newRemotePartySpecialistFloorSeqNumber</code> present	Specify the usage of the <code>SeqNumber</code>
<code>seqNumber</code>	INTEGER	Relevant Originator Counter	OPTIONAL	Specify the associated <code>SeqNumber</code>
<code>otherRemotePartySeqNumberChanges</code>	[6] IMPLICIT SEQUENCE OF		OPTIONAL	In some cases, one party acting in one Remote Party Role may be replacing certificates for a different Remote Party Role (e.g. <code>transitionalCoS</code> changing Supplier Credentials). In such cases, sequence counters need also to be reset for that other Remote Party Role
SEQUENCE				

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
otherRemotePartyRole	INTEGER	supplier (2) , networkOperator (3) , accessControlBroker (4) , transitionalCoS (5) , wanProvider (6) ,	Mandatory if otherRemotePartySeqNumberChanges present	The Remote Party Role of the party whose credentials are being placed on the Device but which didn't authorise the command directly. Note that this is not valid for root or recovery
otherRemotePartyFloorSeqNumber	INTEGER	Relevant Originator Counter	Mandatory if otherRemotePartySeqNumberChanges present	Specify the associated SeqNumber
otherRemotePartySpecialistFloorSeqNumber	SEQUENCE OF		OPTIONAL	Should only be present if otherRemotePartyRole = supplier, and that new supplier uses different counters to prevent replay on Prepayment Top Up
SEQUENCE				
seqNumberUsage	INTEGER	prepaymentTopUp (0)	Mandatory if newRemotePartySpecialistFloorSeqNumber present	Specify the usage of the SeqNumber
seqNumber	INTEGER	Relevant Originator Counter	OPTIONAL	Specify the associated SeqNumber
replacements	SEQUENCE OF			Provide a list of the replacements. Each replacement contains a new

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
				'end entity' Certificate and the identity of the Trust Anchor Cell which is to have its contents replaced using that Certificate.
SEQUENCE			At least one SEQUENCE must be present	One structure is required for each Trust Anchor Cell that is to be updated
replacementCertificate	Certificate	End entity Certificate	Mandatory if SEQUENCE is present	Provide the new end entity certificate
targetTrustAnchorCell	SEQUENCE			Specify where it is to go (specifically which Trust Anchor Cell is to have its details replaced using the new end entity certificate)
trustAnchorCellRemotePartyRole	INTEGER	root (0) , recovery (1) , supplier (2) , networkOperator (3) , accessControlBroker (4) , transitionalCoS (5) , wanProvider (6)	Mandatory if SEQUENCE is present	To which Remote Party Role does the Trust Anchor Cell relate
trustAnchorCellKeyUsage	BIT STRING	{digitalSignature (0) , keyAgreement (4) ,	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
		keyCertSign (5) } ,		
trustAnchorCellUsage	INTEGER	prePaymentTopUp (1) }	DEFAULT management	<p>Should be absent unless:</p> <ul style="list-style-type: none"> 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 Certificate	The list of certificates needed for Certification Path Validation	At least one Certificate must be present since root will never directly sign any end entity certificate so at least Certification Authority certificate is needed	Provide the certificates needed to undertake Certification Path Validation 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

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
			unless replacement is rootBySupplier or rootByWanProvider. In these latter cases a NewWithOld certificate is required for Certification Path Validation, as per section 13.3.5.9.1	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

3869 Table 13.3.4.1: Attribute values for Update Security Credentials Command

3870 **13.3.4.2 Command Authenticity and Integrity Verification**

3871 The Device shall undertake processing according to the requirements of Section 13.3.5.1.

3872 Should any of the checks detailed in Section 13.3.5.1 fail then the Device shall:

- 3873 • generate an entry in the Security Log recording failed Authentication;
- 3874 • discard the Command without execution and without sending a Response; and
- 3875 • send an Alert notifying the failed Authentication, constructed as specified in Section 6.2.4.2 of the GBCS, populated with the relevant
- 3876 Alert Code according to Section 16, to the Known Remote Party identified by:
- 3877 o the Trust Anchor Cell {supplier, digitalSignature, management} if the Device's deviceType is not
- 3878 communicationsHubCommunicationsHubFunction; or

3879 o the Trust Anchor Cell {wanProvider, digitalSignature, management} if the Device's deviceType is
 3880 communicationsHubCommunicationsHubFunction.

3881 Where all of the checks detailed in Section 13.3.5.1 succeed the Device shall process the Command and produce a Response.

3882 13.3.4.3 Command Processing

3883 Before undertaking any further processing, the Device shall update Highest Prior Sequence Number to the value of
 3884 authorisingRemotePartySeqNumber.

3885 If executionDateTime is present then the Device shall:

- 3886 • record against the remotePartyRole (as specified in authorisingRemotePartyControl),
 3887 authorisingRemotePartyControl, replacements; and executionDateTime;
- 3888 • construct a Response where executionOutcome is not present according to the requirements of Section 13.3.4.4; and
- 3889 • at the date-time specified in executionDateTime, undertake the processing of Section 13.3.4.3.1 then construct an Alert according to
 3890 the requirements of Section 13.3.4.5.

3891 If executionDateTime is not present then the Device shall:

- 3892 • undertake the processing of Section 13.3.4.3.1; and
- 3893 • construct a Response where executionOutcome is present according to the requirements of Section 13.3.4.4.

3894 13.3.4.3.1 replacements Processing

3895 For each of the targetTrustAnchorCell in replacements, the Device shall:

- 3896 • record the entityIdentifier and subjectKeyIdentifier currently held in that targetTrustAnchorCell;
- 3897 • attempt to replace the contents of that targetTrustAnchorCell using the corresponding certificate in
 3898 TrustAnchorReplacement; and
- 3899 • if the contents of the replacement are successfully applied, undertake the processing required by Section 13.3.5.10 in relation to the
 3900 RemotePartyRole for that targetTrustAnchorCell.

3901 13.3.4.4 Response Construction

3902 The @UpdateSecurityCredentials.ResponsePayload shall have the structure defined in Section 13.3.5.11, and the Device shall
 3903 populate the executionOutcome, where present with values according to Table 13.3.4.6.

3904 **13.3.4.5 Alert Construction**

3905 The @UpdateSecurityCredentials.AlertPayload shall have the structure defined in Section 13.3.4, and the Device shall populate the
 3906 executionOutcome, with values according to Table 13.3.4.6.

3907 **13.3.4.6 executionOutcome construction**

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
executionOutcome	SEQUENCE			
authorisingRemotePartySeqNumber	INTEGER	Originator Counter of Remote Party authorising the Command, as specified in the corresponding Command	Mandatory	This is to allow the Alert to be linked to the Command that caused execution
credentialsReplacementMode	INTEGER	rootBySupplier (0) , rootByWanProvider (1) , supplierBySupplier (2) , networkOperatorByNetworkOperator (3) , accessControlBrokerByACB (4) , wanProviderByWanProvider (5) , transCoSByTransCoS (6) , supplierByTransCoS (7) ,	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
		anyExceptAbnormalRootByRecovery (8) , anyByContingency (9) } ,		
remotePartySeqNumberChanges	SEQUENCE OF		OPTIONAL	The resulting changes to any replay counters held on the Device
SEQUENCE				
otherRemotePartyRole	INTEGER	root (0) , recovery (1) , supplier (2) , networkOperator (3) , accessControlBreaker (4) , transitionalCoS (5) , wanProvider (6) ,	Mandatory if SEQUENCE is present	The role which has had its counter values changed on the Device
otherRemotePartyFloorSeqNumber	INTEGER	The corresponding Counter value	Mandatory if SEQUENCE is present	
newRemotePartySpecialistFloorSeqNumber	SEQUENCE OF		OPTIONAL	Only present where Remote Party Role is supplier
SEQUENCE				
seqNumberUsage	INTEGER	{prepaymentTopUp (0) } ,	Mandatory if newRemotePartySpecialistFloorSeqN	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
			umber present	
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) ,	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
		keyAgreement (4) , keyCertSign (5)		
trustAnchorCellUsage	INTEGER	{management (0) , prePaymentTopUp (1) }	DEFAULT management	Absent unless: <ul style="list-style-type: none"> 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	ENUMERATE D	success (0) , badCertificate (5) , noTrustAnchor (10) , insufficientMemory (17) , contingencyPublicKeyDecrypt (22) , trustAnchorNotFound (25) , resourcesBusy (30) , other (127)	Mandatory if SEQUENCE is present	Whether the replacement to this Cell was successful or, if it failed, why it failed

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
existingSubjectUniqueID	OCTET STRING		Mandatory if SEQUENCE is present	The 64 bit Entity Identifier of the Remote Party whose credentials were in this Cell prior to receipt of the corresponding Command
existingSubjectKeyIdentifier	OCTET STRING		Mandatory if SEQUENCE is present	For the public key in this Cell prior to receipt of the corresponding Command
replacingSubjectUniqueID	OCTET STRING		Mandatory if SEQUENCE is present	The 64 bit Entity Identifier of the Remote Party whose credentials were to be placed in this Cell
replacingSubjectKeyIdentifier	OCTET STRING		Mandatory if SEQUENCE is present	For the public key which was to be placed in this Cell

3908 Table 13.3.4.6: Attribute values for executionOutcome

3909 13.3.5 Common Requirements

3910 13.3.5.1 Update Security Credentials Command Verification

3911 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
 3912 should be carried out in the order specified. Checking shall cease at the point that any one check fails. The checks required are shown in
 3913 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

Check Number	Criteria that must be tested by the Device	How the Device shall test the Criteria
		CredentialsReplacementMode specified in CommandPayload
1.3	If executionDateTime is present the Command is to replace Supplier Security Credentials.	If executionDateTime is present then credentialsReplacementMode must either supplierBySupplier or supplierByTransCoS
1.4	The Device has not already actioned this Command.	As specified in Section 13.3.5.3
2.1	The targetTrustAnchorCells all exist on a Device of this type	As specified in Section 13.3.5.4
2.2	The credentialsReplacementMode is one that can be Authorised by 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
2.3	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

3914 Table 13.3.5.1: Update Security Credentials Command authenticity and integrity verification

3915 **13.3.5.2 Message Code Validation**

CredentialsReplacementMode	Message Code
rootBySupplier	0x0100
rootByWanProvider	0x0101
supplierBySupplier	0x0102
networkOperatorByNetworkOperator	0x0103
accessControlBrokerByACB	0x0104
wanProviderByWanProvider	0x0105

CredentialsReplacementMode	Message Code
transCoSByTransCoS	0x0106
supplierByTransCoS	0x0107
anyExceptAbnormalRootByRecovery	0x0108
anyByContingency	0x0109

3916 Table 13.3.5.2: Message Code validation against CredentialsReplacementMode

3917 **13.3.5.3 Preventing Replay of Commands**

3918 The Protection Against Replay mechanisms for the Update Security Credentials Command shall be that specified in this Section 13.3.5.3
3919 (which is different than that for other GBCS Commands).

3920 For each of RemotePartyRole from which the Device can receive a valid Updated Security Credentials Command, the Device shall allocate
3921 storage for a Highest Prior Sequence Number which shall be capable of storing a 64 bit unsigned integer and which shall initially be set to the
3922 value zero at manufacture.

3923 Before executing any Update Security Credentials Command, a Device shall confirm that, if CredentialsReplacementMode <>
3924 accessControlBrokerByACB, then

- 3925 • (authorisingRemotePartyTACellIdentifier is populated in the Command) and (the authorisingRemotePartySeqNumber
3926 is strictly numerically greater than the Highest Prior Sequence Number the Device has recorded for the RemotePartyRole identified in
3927 authorisingRemotePartyTACellIdentifier)

3928 else

- 3929 • (the authorisingRemotePartySeqNumber is strictly numerically greater than the Highest Prior Sequence Number the Device has
3930 recorded for the accessControlBroker)

3931 **13.3.5.4 Required Trust Anchor Cells by Device Type**

3932 The Trust Anchor Cells specified in Section 4.3.2.5 are those required on each Device type and so are the only valid
3933 targetTrustAnchorCells.

3934 The Device shall ensure that all targetTrustAnchorCells specified in the Command instance are valid for the type of Device it is,
3935 according to the requirements of Section 4.3.2.5. A Command containing any invalid targetTrustAnchorCells shall not be processed any
3936 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.

	<i>authorisingRemotePartyTACellIdentifier</i>		
	<i>RemotePartyRole</i>	<i>KeyUsage</i>	<i>CellUsage</i>
<i>credentialsReplacementMode</i>			
<i>rootBySupplier</i>	<i>supplier</i>	<i>digitalSignature</i>	<i>management</i>
<i>rootByWanProvider</i>	<i>wanProvider</i>	<i>digitalSignature</i>	<i>management</i>
<i>supplierBySupplier</i>	<i>supplier</i>	<i>digitalSignature</i>	<i>management</i>
<i>networkOperatorByNetworkOperator</i>	<i>networkOperator</i>	<i>digitalSignature</i>	<i>management</i>
<i>accessControlBrokerByACB</i> : if <i>authorisingRemotePartyTACellIdentifier</i> is present it must refer to the specified Trust Anchor Cell. If absent and <i>deviceType</i> is not <i>communicationsHubCommunicationsHubFunction</i> , then the mode of <i>accessControlBrokerByACB</i> is valid, otherwise the mode is invalid	<i>accessControlBroker</i>	<i>digitalSignature</i>	<i>management</i>
<i>wanProviderByWanProvider</i>	<i>wanProvider</i>	<i>digitalSignature</i>	<i>management</i>
<i>transCoSByTransCoS</i>	<i>transitionalCoS</i>	<i>digitalSignature</i>	<i>management</i>
<i>supplierByTransCoS</i>	<i>transitionalCoS</i>	<i>digitalSignature</i>	<i>management</i>
<i>anyExceptAbnormalRootByRecovery</i>	<i>recovery</i>	<i>digitalSignature</i>	<i>management</i>
<i>anyByContingency</i>	<i>root</i>	<i>keyCertSign</i>	<i>management</i>

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.

3948 Table 13.3.5.6 below lists the only valid `targetTrustAnchorCell` combinations for each `credentialsReplacementMode`. All other
3949 combinations are invalid. A Command containing any invalid combinations shall not be processed any further by the Device.

	<code>targetTrustAnchorCell</code>		
	<code>RemotePartyRole</code>	<code>KeyUsage</code>	<code>CellUsage</code>
<code>credentialsReplacementMode</code>			
<code>rootBySupplier</code>	<code>root</code>	<code>keyCertSign</code>	<code>management</code>
<code>rootByWanProvider</code>	<code>root</code>	<code>keyCertSign</code>	<code>management</code>
<code>supplierBySupplier</code>	<code>supplier</code>	any valid for GBCS	any valid for GBCS
<code>networkOperatorByNetworkOperator</code>	<code>networkOperator</code>	any valid for GBCS	any valid for GBCS
<code>accessControlBrokerByACB</code>	<code>accessControlBroker</code>	any valid for GBCS	any valid for GBCS
<code>wanProviderByWanProvider</code>	<code>wanProvider</code>	any valid for GBCS	any valid for GBCS
<code>transCoSByTransCoS</code>	<code>transitionalCoS</code>	<code>digitalSignature</code>	<code>management</code>
<code>supplierByTransCoS</code>	<code>supplier</code>	any valid for GBCS	any valid for GBCS
<code>anyExceptAbnormalRootByRecovery</code>	any valid for GBCS	any valid for GBCS	any valid for GBCS
<code>anyByContingency</code>	any valid for GBCS	any valid for GBCS	any valid for GBCS

3950 Table 13.3.5.6: Valid `credentialsReplacementMode` by the `targetTrustAnchorCells` specified in the Command

3951 *13.3.5.7 Valid usage of Certificates against the `targetTrustAnchorCells` specified in the Command*

3952 [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.]

3953 For each instance of the `TrustAnchorReplacement` structure in the Command, the `keyUsage` in `replacementCertificate` shall be
3954 equal to `targetTrustAnchorCell.KeyUsage`. Where this check fails for any one or more of the `TrustAnchorReplacement` instances,
3955 the Command shall not be actioned by the Device.

3956 [Note: Save for supplier and network operator roles, each of the ‘end entity’ Certificates in the Command must have the same
3957 `RemotePartyRole` as the Trust Anchor Cell it is to be applied to.]

3958 For each instance of the `TrustAnchorReplacement` structure in the Command where (`targetTrustAnchorCell.RemotePartyRole` <>
3959 `supplier`) and (`targetTrustAnchorCell.RemotePartyRole` <> `networkOperator`), the `RemotePartyRole` in
3960 `replacementCertificate` shall be equal to `targetTrustAnchorCell.RemotePartyRole`. Where this check fails for any one or more
3961 of the `TrustAnchorReplacement` instances, the Command shall not be actioned by the Device.

3962 Notes:

- 3963 • mismatches between RemotePartyRole in the certificate and the target Trust Anchor Cell are admissible for supplier and
- 3964 networkOperator only, and are needed (see Section 4.3.2.5); and
- 3965 • CellUsage is simply a selector to allow a different Key Agreement key pair to be used for Prepayment Top Ups. However, that use of
- 3966 a different Key Pair is not mandated and so validation is not required; any valid supplier Key Agreement certificate may be used in this
- 3967 Trust Anchor Cell.

3968 *13.3.5.8 Verifying the Cryptographic Protections*

3969 In verifying Cryptographic Protections pursuant to this Section 13.3.5.8.1:

- 3970 • KRP Signature shall be verified according to the requirements in Section 4.3.2.7.2; and
- 3971 • ACB-SMD MAC shall be verified according to the requirements in Section 6.2.4.1.2.

3972 If credentialsReplacementMode = anyByContingency then KRP Signature shall be verified using the public key established according

3973 to the requirements of Section 13.3.5.8.1.

3974 If credentialsReplacementMode = <> anyByContingency then KRP Signature shall be verified using the public key identified as per

3975 Section 4.3.2.7.2.

3976 If credentialsReplacementMode = accessControlBrokerByACB and deviceType is not

3977 communicationsHubCommunicationsHubFunction then ACB-SMD MAC shall be verified as per Section 6.2.4.1.2.

3978 *13.3.5.8.1 Decrypting the contingency public key and verifying Authorising Remote Party's digital signature against that decrypted key*

3979 The Device shall decrypt the Contingency Key that it holds in Trust Anchor Cell {root, digitalSignature, management} by undertaking

3980 Decryption using the following parameters:

- 3981 • setting Ciphertext to be encrypted value of the Contingency Key;
- 3982 • setting Additional Authenticated Data to be 0x31;
- 3983 • setting the Initialization Vector to be 0xFFFFFFFF0000000000000000; and
- 3984 • setting the shared symmetric key to be the value in plaintextSymmetricKey.

3985 Where Decryption is successful, the Device shall use the Plaintext produced as the Public Key to verify KRP Signature according to the

3986 requirements at Section 6.3.4.

3987 The Contingency Key shall have been Encrypted accordingly.

3988 **13.3.5.9 Verifying the authenticity of replacement certificates**

3989 The Device shall first apply the requirements of Section 12.6 (Device processing of Certificates). If any of those checks fail, the Section
3990 13.3.5.9.1 check fails.

3991 Where Certification Path Validation is required by this Section 13.3.5.9, the application of time based checks shall be determined as follows:

- 3992 • if, in the Command, `applyTimeBasedCPVChecks = disapply` then time based checks shall NOT be applied by the Device;
- 3993 • otherwise time based checks shall be applied or not applied in line with the requirements of Section 4.3.2.8.

3994 If (`credentialsReplacementMode <> anyByContingency`) and (`replacements` does NOT include a `targetTrustAnchorCell` of
3995 `{root, keyCertSign}`) then the Device shall, for each `replacementCertificate` in `replacements`, undertake Certification Path
3996 Validation according to the requirements of Section 4.3.2.8.

3997 If (`credentialsReplacementMode <> anyByContingency`) and (`replacements` does include a `targetTrustAnchorCell` of `{root,`
3998 `keyCertSign}`) then the Device shall first undertake the checks at Section 13.3.5.9.1 in relation to the `root` Certificates and then shall, for
3999 each of the other `replacementCertificate` in `replacements`, undertake Certification Path Validation according to the requirements of
4000 Section 4.3.2.8.

4001 If (`credentialsReplacementMode = anyByContingency`) and (`replacements` does include a `targetTrustAnchorCell` of `{root,`
4002 `keyCertSign}`) then the Device shall, for each of the other `replacementCertificate` in `replacements`, undertake Certification Path
4003 Validation according to the requirements of Section 4.3.2.8. In so doing the Device shall use the details from the `replacementCertificate`
4004 in `replacements` specified for updating `{root, keyCertSign}` as the root for Certification Path Validation.

4005 **13.3.5.9.1 Validation of new root Certificate against current root Security Credentials**

4006 The Device shall:

- 4007 • identify the Certificate in `replacements` that corresponds to the `targetTrustAnchorCell` of `{root, keyCertSign}`. The
4008 Certificate shall be referred to as `NewWithNew`; then
- 4009 • identify the Certificate in `certificationPathCertificates` that has the same `subjectKeyIdentifier` as the
4010 `NewWithNew` Certificate. The Certificate shall be referred to as `NewWithOld`. If no such Certificate is found, the Section
4011 13.3.5.9.1 check fails else:
- 4012 • undertake Certification Path Validation on `NewWithOld` according to the requirements of Section 4.3.2.8. If the Certification Path
4013 Validation fails, the Section 13.3.5.9.1 check fails else:

- 4014 • use the Public Key in `NewWithNew` to verify the digital signature in `NewWithNew`. If the digital signature verification fails, the Section
4015 13.3.5.9.1 check fails.
- 4016 **13.3.5.10 Required Processing on Change of Remote Party Control**
- 4017 If:
- 4018 • the `targetTrustAnchorCell` is `{supplier, digitalSignature, management}`; and
 - 4019 • the Entity Identifier in the `targetTrustAnchorCell` is changed by the replacement; and
 - 4020 • the Device is an ESME or a GSME,
- 4021 then the Device shall:
- 4022 • set the Supplier Name which it displays to the X.500 Distinguished Name in the `subject` field of the `certificate` that was used
4023 to populate the `targetTrustAnchorCell`; and
 - 4024 • add an entry in the Billing Data Log with a snapshot cause of 0x00020000 (Change of Supplier) (with the entry added having the same
4025 content as is required on Set Payment Mode Or Tariff change); and
 - 4026 • reset the Tariff Block Counter Matrix.
- 4027 If the `targetTrustAnchorCell` is `{root, keyCertSign, management}` and there are any future dated Update Security Credentials or
4028 Activate Firmware Commands held on the Device that have not yet executed, and so the `executionDateTime` is in the future, then the
4029 Device shall set each `executionDateTime` to '99991231235959Z'.
- 4030 If the `targetTrustAnchorCell` is not `{root, keyCertSign, management}` and there are any future dated Update Security
4031 Credentials or Activate Firmware Commands held on the Device that:
- 4032 • include replacements for this `remotePartyRole`; and
 - 4033 • have not yet executed, and so the `executionDateTime` is in the future:
- 4034 then the Device shall set each `executionDateTime` to '99991231235959Z'.
- 4035 If:
- 4036 • the `targetTrustAnchorCell` is `{supplier, digitalSignature, management}` or `{root, keyCertSign,`
4037 `management}`; and
 - 4038 • the Entity Identifier in the `targetTrustAnchorCell` is changed by the replacement

4039 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
 4040 '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
 4041 are pending activation.

4042 If:

- 4043 • `remotePartyRole` of `targetTrustAnchorCell` and that of `authorisingRemotePartyControl` is `supplier`; and
- 4044 • `keyUsage` of `targetTrustAnchorCell` is `digitalSignature`

4045 then the Device shall:

- 4046 • set all Execution Counters to the value in `newRemotePartyFloorSeqNumber`;
- 4047 • clear all values from the UTRN Counter Cache; and
- 4048 • place a single value in the UTRN Counter Cache. If `newRemotePartySpecialistFloorSeqNumber` is present and the
 4049 `seqNumberUsage` in that `newRemotePartySpecialistFloorSeqNumber` is `prepaymentTopUp` then that value shall be the 32
 4050 most significant bits of the `seqNumber` in the `newRemotePartySpecialistFloorSeqNumber`. Otherwise the value shall be the 32
 4051 most significant bits of the `newRemotePartyFloorSeqNumber`.

4052 If (`remotePartyRole` of `authorisingRemotePartyControl` is not `supplier`) but (`targetTrustAnchorCell` is {`supplier`,
 4053 `digitalSignature`, `management`}) then there should be an instance of `otherRemotePartySeqNumberChanges` where
 4054 `remotePartyRole` is `supplier` in the Command. Using the values in that instance of `otherRemotePartySeqNumberChanges` or the
 4055 values zero if there is no such instance, the Device shall:

- 4056 • set all Execution Counters to the value in `otherRemotePartyFloorSeqNumber`;
- 4057 • clear all values from the UTRN Counter Cache; and
- 4058 • place a single value in the UTRN Counter Cache. If `newRemotePartySpecialistFloorSeqNumber` is present and the
 4059 `seqNumberUsage` in that `newRemotePartySpecialistFloorSeqNumber` is `prepaymentTopUp` then that value shall be the 32
 4060 most significant bits of the `seqNumber` in the `newRemotePartySpecialistFloorSeqNumber`. Otherwise the value shall be the 32
 4061 most significant bits of the `otherRemotePartyFloorSeqNumber`.

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
{
```

```

4102      -- if the Command is future dated, the Response will not have any details of execution (those will be in the subsequent alert)
4103
4104      commandAccepted                      NULL,
4105
4106      -- if the Command is for immediate execution, the Response will detail the outcomes
4107
4108      executionOutcome                     ExecutionOutcome OPTIONAL
4109
4110  }
4111
4112  AlertPayload ::=                         SEQUENCE
4113  {
4114      -- specify the Alert Code
4115      alertCode                           INTEGER(0..4294967295),
4116
4117      -- specify the date-time of execution
4118      executionDateTime                    GeneralizedTime,
4119
4120
4121      -- detail what happened when the future dated Command was executed
4122
4123      executionOutcome                     ExecutionOutcome
4124  }
4125
4126  ExecutionOutcome ::=                     SEQUENCE
4127  {
4128      -- Provide details of the corresponding Command that may not be in the standard GBCS message header. Specifically the
4129      -- mode in which the Command was invoked, the Originator Counter in the original Command and the resulting changes to any
4130      -- replay counters held on the Device
4131
4132      authorisingRemotePartySeqNumber      SeqNumber,
4133      credentialsReplacementMode           CredentialsReplacementMode,
4134      remotePartySeqNumberChanges          SEQUENCE OF RemotePartySeqNumberChange,
4135
4136      -- For each replacement in the Command, detail the outcome and impacted parties
4137
4138      replacementOutcomes                   SEQUENCE OF ReplacementOutcome
4139  }
4140
4141
4142  AuthorisingRemotePartyControl ::=        SEQUENCE
4143  {
4144      -- Specify the replacement mode so that the Device can check that the Remote Party Role is allowed to

```

```

4146 -- authorise this type of replacement and that all replacements in the payload are allowed within this
4147 -- replacement mode
4148
4149 credentialsReplacementMode          CredentialsReplacementMode,
4150
4151 -- Only if credentialsReplacementMode = anyByContingency, provide the symmetric key to decrypt
4152 -- the Contingency Public Key in the (root, digitalSignature, management) Trust Anchor Cell
4153
4154 plaintextSymmetricKey                [0] IMPLICIT OCTET STRING OPTIONAL,
4155
4156 -- Specify whether the time based checks as part of any Certificate Path Validation should be applied
4157
4158 applyTimeBasedCPVChecks              [1] IMPLICIT INTEGER {apply(0), disapply(1)} DEFAULT apply,
4159
4160 -- Identify which of the Public Keys on the Device is to be used in checking KRP Signature
4161 -- 'authorisingRemotePartyTACellIdentifier' may only be omitted when
4162 -- the access control broker is updating its own credentials and the target device is not a CHF.
4163 -- In all other cases it is mandatory.
4164
4165 authorisingRemotePartyTACellIdentifier [2] IMPLICIT TrustAnchorCellIdentifier OPTIONAL,
4166
4167 -- Specify the Originator Counter for the Remote Party Applying KRP Signature, or (for the
4168 -- Access Control Broker changing its credentials) the Access Control Broker's Originator Counter.
4169
4170 authorisingRemotePartySeqNumber       [3] IMPLICIT SeqNumber,
4171
4172 -- If the Command is to effect a change of control, then newTrustAnchorFloorSeqNumber must be included
4173 -- and will be the value used to prevent replay of Update Security Credentials Commands for the
4174 -- new controlling Remote Party.
4175
4176 newRemotePartyFloorSeqNumber          [4] IMPLICIT SeqNumber OPTIONAL,
4177
4178 -- Some Commands on the Device may use a different Originator Counter sequence for Protection Against Replay. At this
4179 -- version of the GBCS, the only example is the Prepayment Top Up Command on ESME and GSME. The
4180 -- SpecialistSeqNumber structure allows such Counters to also be reset on change of control.
4181
4182 newRemotePartySpecialistFloorSeqNumber [5] IMPLICIT SEQUENCE OF SpecialistSeqNumber OPTIONAL,
4183
4184 -- In some cases, one party acting in one Remote Party Role may be replacing certificates for a different Remote Party Role.
4185 -- In some cases, sequence counters need also to be reset for those other Remote Party Role(s)
4186
4187 otherRemotePartySeqNumberChanges      [6] IMPLICIT SEQUENCE OF RemotePartySeqNumberChange OPTIONAL
4188 }
4189

```

```

4190 RemotePartySeqNumberChange ::=          SEQUENCE
4191 {
4192   otherRemotePartyRole                RemotePartyRole,
4193   otherRemotePartyFloorSeqNumber      SeqNumber,
4194   newRemotePartySpecialistFloorSeqNumber SEQUENCE OF SpecialistSeqNumber OPTIONAL
4195 }
4196
4197 SpecialistSeqNumber ::=                  SEQUENCE
4198 {
4199   -- Specify the usage of the SeqNumber
4200   seqNumberUsage                      SeqNumberUsage,
4201
4202   -- Specify the associated SeqNumber
4203   seqNumber                           SeqNumber
4204 }
4205
4206 SeqNumberUsage ::=                      INTEGER
4207 {
4208   -- Define the full set of discrete usages on a Device. The only specialist
4209   -- counter is for Prepayment Top Up (which is set independently of other counters). This may only be
4210   -- included when changing Supplier Security Credentials on an ESME or GSME.
4211
4212   prepaymentTopUp                      (0)
4213 }
4214
4215 SeqNumber ::=                          INTEGER (0.. 18446744073709551615)
4216
4217
4218 TrustAnchorReplacement ::=              SEQUENCE
4219 {
4220   -- Provide the new end entity certificate
4221
4222   replacementCertificate                Certificate,
4223
4224   -- Specify where it is to go (specifically which Trust Anchor Cell is to have its details replaced using
4225   -- the new end entity certificate)
4226
4227   targetTrustAnchorCell                 TrustAnchorCellIdentifier
4228 }
4229
4230
4231 ReplacementOutcome ::=                  SEQUENCE
4232 {
4233   affectedTrustAnchorCell                TrustAnchorCellIdentifier,

```

```

4234      statusCode                      StatusCode,
4235
4236      -- The GBCS Certificate requirements mean that the subjectUniqueID attribute in the subject field of a certificate will always
4237      -- contain the 64 bit unique number that equates to Entity Identifier. existingSubjectUniqueID should be set
4238      -- accordingly based on the contents of the Trust Anchor Cell prior to Command processing.
4239
4240      existingSubjectUniqueID          OCTET STRING,
4241
4242      -- The GBCS Certificate requirements mean that subjectKeyIdentifier attributes will all be 8 byte SHA-1 Hashes.
4243      -- existingSubjectKeyIdentifier should be set accordingly based on the contents of the Trust Anchor Cell prior to
4244      -- Command processing.
4245
4246      existingSubjectKeyIdentifier      OCTET STRING,
4247
4248      -- The subjectUniqueID in the subject field of the certificate in this TrustAnchorReplacement
4249
4250      replacingSubjectUniqueID          OCTET STRING,
4251
4252      -- The subjectKeyIdentifier in the certificate in this TrustAnchorReplacement
4253
4254      replacingSubjectKeyIdentifier      OCTET STRING
4255  }
4256
4257  TrustAnchorCellIdentifier ::=          SEQUENCE
4258  {
4259      -- Which Remote Party Role does this Cell relate to?
4260
4261      trustAnchorCellRemotePartyRole    RemotePartyRole,
4262
4263      -- To what cryptographic use can the Public Key in this Cell be put? Some Remote Party Roles
4264      -- (e.g. supplier) can have more than one Public Key on a Device and each one would only have
4265      -- a single cryptographic use.
4266
4267      trustAnchorCellKeyUsage           KeyUsage,
4268
4269      -- trustAnchorCellUsage is to allow for multiple Public Keys of the same keyUsage for the same Remote
4270      -- Party Role. It will be absent except where used to refer to the Supplier Key
4271      -- Agreement Key used solely in relation to validating Supplier generated MACs on Prepayment Top Up
4272      -- transactions
4273
4274      trustAnchorCellUsage              CellUsage DEFAULT management
4275  }
4276
4277  CellUsage ::=
4278      INTEGER {management(0), prePaymentTopUp(1)}

```

```

4278
4279 RemotePartyRole ::=                                INTEGER
4280 {
4281 -- Define the full set of Remote Party Roles in relation to which a Device may need to undertake
4282 -- processing. Note that most Devices will only support a subset of these.
4283
4284 root                                (0),
4285 recovery                            (1),
4286 supplier                            (2),
4287 networkOperator                    (3),
4288 accessControlBroker                (4),
4289 transitionalCoS                    (5),
4290 wanProvider                        (6),
4291 issuingAuthority                    (7),    -- Devices will receive such Certificates but they do not need to store
4292                                         -- them over an extended period
4293
4294 -- The 'other' RemotePartyRole is for a party whose role does not allow it to invoke any Device function apart from
4295 -- UpdateSecurityCredentials. This is to allow for Device functionality to be locked out of usage until a valid
4296 -- Remote Party can be identified e.g. where roles cannot be fixed until a Device is brought in to operation
4297
4298 other                                (127)
4299
4300 }
4301
4302 -- KeyUsage is only repeated here for clarity. It is defined in RFC 5912
4303
4304 KeyUsage ::=                                        BIT STRING
4305 {
4306 -- Define valid uses of Public Keys held by Devices in their Trust Anchor Cells.
4307
4308 digitalSignature                    (0),
4309 contentCommitment                  (1),    -- not valid for GBCS compliant transactions
4310 keyEncipherment                    (2),    -- not valid for GBCS compliant transactions
4311 dataEncipherment                  (3),    -- not valid for GBCS compliant transactions
4312 keyAgreement                      (4),
4313 keyCertSign                       (5),
4314 cRLSign                          (6),
4315 encipherOnly                      (7),    -- not valid for GBCS compliant transactions
4316 decipherOnly                      (8)    -- not valid for GBCS compliant transactions
4317 }
4318
4319 CredentialsReplacementMode ::=                INTEGER
4320 {
4321 -- Define the valid combinations as to which Remote Party Roles can replace which kinds of Trust Anchors.

```



```

4322
4323 -- Normal operational replacement modes
4324 rootBySupplier (0),
4325 rootByWanProvider (1),
4326 supplierBySupplier (2),
4327 networkOperatorByNetworkOperator (3),
4328 accessControlBrokerByACB (4),
4329 wanProviderByWanProvider (5),
4330 transCoSByTransCoS (6),
4331 supplierByTransCoS (7),
4332
4333 -- Recovery modes
4334 anyExceptAbnormalRootByRecovery (8),
4335 anyByContingency (9)
4336 }
4337
4338 -- The GBCS only allows for a constrained set of Trust Anchor Cell operations and so the list of possible outcomes
4339 -- is more limited than in RFC 5934. The list below is that more constrained subset
4340
4341 StatusCode ::= ENUMERATED {
4342
4343 success (0),
4344
4345 -- badCertificate is used to indicate that the syntax for one or more certificates is invalid.
4346
4347 badCertificate (5),
4348
4349 -- noTrustAnchor is used to indicate that the authorityKeyIdentifier does not identify the public key of a
4350 -- trust anchor or a certification path that terminates with an installed trust anchor
4351
4352 noTrustAnchor (10),
4353
4354 -- insufficientMemory indicates that the update could not be processed because the Device did not
4355 -- have sufficient memory
4356
4357 insufficientMemory (17),
4358
4359 -- contingencyPublicKeyDecrypt indicates that the update could not be processed because an error occurred while
4360 -- decrypting the contingency public key.
4361
4362 contingencyPublicKeyDecrypt (22),
4363
4364 -- trustAnchorNotFound indicates that a change to a trust anchor was requested, but the referenced trust anchor
4365 -- is not represented in the Trust Anchor Cell.

```

```

4366
4367 trustAnchorNotFound                (25),
4368
4369 -- resourcesBusy indicates that the resources necessary to process the replacement are not available at the
4370 -- present time, but the resources might be available at some point in the future.
4371
4372 resourcesBusy                        (30),
4373
4374 -- other indicates that the update could not be processed, but the reason is not covered by any of the assigned
4375 -- status codes. Use of this status code SHOULD be avoided.
4376
4377 other                                (127) }
4378
4379 END

```

4380 **13.3.5.12 Requirements for AuthorisingRemotePartyControl elements – informative**

4381 All bar two parts of the AuthorisingRemotePartyControl structure are optional. This section summarises when each of the optional
 4382 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 newTrustAnchorFloorSeqNumber should be included. It can be present in all other situations
newRemotePartySpecialistFloorSeqNumber	Only required on Change of Supplier where the new Supplier has decided to use a different sequence of Originator Counters for prepayment top ups.

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 Protection Against Replay counters for that other role or roles

4383 Table 13.3.5.12: Requirements for AuthorisingRemotePartyControl elements

13.4 CS02c Issue Security Credentials

13.4.1 Description

This section covers the creation, validation and processing of (i) Issue Security Credentials Commands and (ii) Responses to such Commands.

13.4.2 Use Case Cross References

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Command and Response
Message Type Category	SME.C.C
Capable of future dated invocation?	No
Protection Against Replay Required?	Yes
SEC User Interface Services Schedule (Service Request) Reference	6.17
Valid Target Device(s)	ESME / GSME / GPF / CHF
Valid Business Originator role(s) for Command invocation (and so, for DLMS COSEM Commands, which Application Association is to be used for delivery of the Command to the Device) [Remote Party Messages Only]	Supplier (for Devices other than CHF) WAN Provider (for CHF Devices only)
Valid Response Recipient role(s) (only for Messages Authorised by the Access Control Broker on behalf of parties not known to the Device) [Remote Party Messages Only]	None
Valid initiating Device type(s) [HAN Only Messages]	None
Protocol	ASN.1

Table 13.4.2: Use Case Cross References for Issue Security Credential Details Command and Response

13.4.3 Construction of Commands

Issue Security Credentials Command Payloads shall be constructed as specified in Section 13.4.7 and Cryptographic Protection I and Cryptographic Protection II shall be applied as required for a Command of Message Category SME.C.C.

13.4.4 Device processing of Commands and Response handling

The Device receiving an Issue Security Credentials Command shall undertake processing steps in the sequence defined in this Section 13.4.4.

In processing an Issue Security Credentials Command, the Device shall:

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:
 - o those held in the {wANProvider, digitalSignature, management} Trust Anchor Cell, if the target Device's deviceType equals communicationsHubCommunicationsHubFunction;

- 4405 ○ those held in the {supplier, digitalSignature, management} Trust
 4406 Anchor Cell, if the target Device's deviceType does not equal
 4407 communicationsHubCommunicationsHubFunction;
- 4408 2. validate that the value of keyUsage in CommandPayload is either
 4409 digitalSignature only or keyAgreement only. If this validation fails then the
 4410 Device shall set issueCredentialsResponseCode to invalidKeyUsage, and
 4411 process from step 6;
- 4412 3. generate a Private-Public Key Pair and store the Private Key so generated in the
 4413 Pending Private Key Cell determined by the value of keyUsage in CommandPayload.
 4414 If the step fails then the Device shall set issueCredentialsResponseCode to
 4415 keyPairGenerationFailed, and process from step 6;
- 4416 4. with the ASN.1 terms in this step (that are not defined in this Section 13.4.4) having the
 4417 meaning of IETF RFC 2986³⁰; generate a CertificationRequest which:
- 4418 ○ complies with the requirements of IETF RFC2986 and IETF RFC 5912;
 4419 ○ is DER encoded, in line with the recommendation of IETF RFC 5967³¹;
 4420 ○ has subjectPublicKey set to the bit string representation of the Public Key
 4421 generated in step 3;
 4422 ○ incorporates an extensionRequest identified by id-ce-keyUsage which shall
 4423 contain the keyUsage value specified in CommandPayload;
 4424 ○ incorporates an extensionRequest identified by id-ce-subjectAltName
 4425 which shall contain a single GeneralName of type OtherName that is further sub-
 4426 typed as a HardwareModuleName (id-on-HardwareModuleName) as defined
 4427 in IETF RFC 4108. The hwSerialNum field shall be set to the Device's Entity
 4428 Identifier; and
 4429 ○ has the signature generated using the Private Key generated in step 3;
- 4430 5. if the generation of CertificationRequest is not successful then the Device shall
 4431 set issueCredentialsResponseCode to cRProductionFailed;
- 4432 6. create a Response according to the requirements of Section 13.4.7, apply the Response
 4433 Cryptographic Protection required for a Response of Message Category SME.C.C, and
 4434 send the Response.

4435 13.4.5 Response Processing

4436 Response Recipient Verification may be undertaken as specified in this GBCS for a
 4437 Response of Message Category SME.C.C. The issueCredentialsResponseCode
 4438 field, where present in the Response, may be decoded according to the ASN.1 definitions at
 4439 Section 13.4.6.

³⁰ <https://tools.ietf.org/html/rfc2986>

³¹ <https://tools.ietf.org/html/rfc5967>

4440 13.4.6 Issue Security Credentials Command and Response payloads – structure definition

4441 Each instance of @IssueSecurityCredentials.CommandPayload and of @IssueSecurityCredentials.ResponsePayload shall be
 4442 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
 4443 notation.

```

4444 IssueSecurityCredentials DEFINITIONS ::= BEGIN
4445
4446 CommandPayload ::=                               SEQUENCE
4447 {
4448     -- specify the keyUsage to which the generated key-pair will be put, if subsequently authorised
4449     keyUsage                               KeyUsage
4450 }
4451
4452 ResponsePayload ::=                               CHOICE
4453 {
4454     -- if the Command was successful, provide the generated Certification Request. CertificationRequest shall
4455     -- be as defined in ASN.1 by IETF RFC 5912. For reference, it is in the section headed 'ASN.1 Module for RFC 2986'
4456     certificationRequest                    CertificationRequest,
4457
4458     -- if the Command was unsuccessful, detail the failure
4459
4460     issueCredentialsResponseCode            IssueCredentialsResponseCode
4461 }
4462
4463 -- KeyUsage is only repeated here for ease of reference. It is defined in IETF RFC 5912
4464
4465 KeyUsage ::=                                     BIT STRING
4466 {
4467     -- Define valid uses of Public Keys held by Devices in their Trust Anchor Cells.
4468
4469     digitalSignature                        (0),
4470     contentCommitment                      (1), -- not valid for GBCS compliant transactions
4471     keyEncipherment                       (2), -- not valid for GBCS compliant transactions
4472     dataEncipherment                      (3), -- not valid for GBCS compliant transactions
4473     keyAgreement                          (4),
4474     keyCertSign                           (5), -- not valid for this Use Case
4475     cRLSign                               (6), -- not valid for this Use Case
4476     encipherOnly                          (7), -- not valid for GBCS compliant transactions
4477     decipherOnly                          (8)  -- not valid for GBCS compliant transactions
4478 }
4479
4480

```

```
4481
4482 IssueCredentialsResponseCode::=          INTEGER
4483 {
4484     invalidKeyUsage                      (1),
4485     keyPairGenerationFailed             (2),
4486     cRProductionFailed                  (3)
4487 }
4488
4489
4490 END
```

4491 **13.4.7 Constructing the @IssueSecurityCredentials.CommandPayload and of**
4492 **@IssueSecurityCredentials.ResponsePayload**

4493 @IssueSecurityCredentials.CommandPayload shall have the structure defined in Section 13.4.6, and the Remote Party constructing
4494 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

4495 Table 13.4.7a: @IssueSecurityCredentials.CommandPayload population

4496 @IssueSecurityCredentials.ResponsePayload shall have the structure defined in Section 13.4.6, and the Device constructing the
4497 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			
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

4498 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
SEC User Interface Services Schedule (Service Request) Reference	6.15
Valid Target Device(s)	ESME / GSME / GPF / CHF
Valid Business Originator role(s) for Command invocation (and so, for DLMS COSEM Commands, which Application Association is to be used for delivery of the Command to the Device) [Remote Party Messages Only]	Supplier (for Devices other than CHF) WAN Provider (for CHF Devices only)
Valid Response Recipient role(s) (only for Messages Authorised by the Access Control Broker on behalf of parties not known to the Device) [Remote Party Messages Only]	None
Valid initiating Device type(s) [HAN Only Messages]	None
Protocol	ASN.1

Table 13.5.2: Use Case Cross References for Update Device Certificate on Device, Command and Response

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. The Security Credentials used to verify Cryptographic Protection I shall be:
 - o those held in the {wANProvider, digitalSignature, management} Trust Anchor Cell, if the target Device's deviceType equals communicationsHubCommunicationsHubFunction; or

- 4520 o those held in the {supplier, digitalSignature, management} Trust
 4521 Anchor Cell, if the target Device's deviceType does not equal
 4522 communicationsHubCommunicationsHubFunction.
- 4523 2. establish the values of keyUsage, subjectPublicKey and hwSerialNum in
 4524 certificate in the CommandPayload. If any of the values cannot be established then the
 4525 Device shall set updateDeviceCertResponseCode to invalidCertificate, and
 4526 process from step 10;
- 4527 3. validate that hwSerialNum established at step 2 is the Device's Entity Identifier. If this
 4528 validation fails then the Device shall set updateDeviceCertResponseCode to
 4529 wrongDeviceIdentity, and process from step 10;
- 4530 4. validate that keyUsage established at step 2 is either digitalSignature only or
 4531 keyAgreement only. If this validation fails then the Device shall set
 4532 updateDeviceCertResponseCode to invalidKeyUsage, and process from step
 4533 10;
- 4534 5. validate that the Device holds a Pending Private Key for the keyUsage as established
 4535 at step 2. If this validation fails then the Device shall set
 4536 updateDeviceCertResponseCode to noCorrespondingKeyPairGenerated, and
 4537 process from step 10;
- 4538 6. validate that subjectPublicKey established at step 2 is the bit string representation
 4539 of the Public Key corresponding to the Pending Private Key identified at step 5. If this
 4540 validation fails then the Device shall set updateDeviceCertResponseCode to
 4541 wrongPublicKey, and process from step 10;
- 4542 7. store certificate. If this step fails then the Device shall set
 4543 updateDeviceCertResponseCode to certificateStorageFailed, and process
 4544 from step 10;
- 4545 8. set the Current Private Key to have the value of the Pending Private Key for the
 4546 keyUsage established at step 2. If this step fails then the Device shall set
 4547 updateDeviceCertResponseCode to privateKeyChangeFailed, and process
 4548 from step 10;
- 4549 9. set updateDeviceCertResponseCode to success; and
- 4550 10. create a Response according to the requirements of Section 13.5.7, apply the Response
 4551 Cryptographic Protection required for a Response of Message Category SME.C.C, and
 4552 send the Response.

4553 If all steps were successful and this was a change of digitalSignature certificate,
 4554 the Response shall be signed using the private key corresponding to the new
 4555 certificate. If there was a failure, the Response shall be signed using the private key
 4556 corresponding to the pre-existing key pair.

4557 Once the Pending Private Key becomes the Current Private Key, the Device will be using
 4558 the new Private Key and this will affect all Remote Parties interacting with the Device;
 4559 specifically they will need to use the new Certificate corresponding to the Private Key now in
 4560 use.

4561 13.5.5 Response Processing

4562 Response Recipient Verification may be undertaken as specified in this GBCS for a
 4563 Response of the relevant Message Category. The updateDeviceCertResponseCode
 4564 field may be decoded according to the ASN.1 definitions at Section 13.5.6.

4565 If this was a change of `digitalSignature` certificate, the public key to be used to verify
4566 the Device's signature is dependent on the value of `updateDeviceCertResponseCode`.
4567 If `updateDeviceCertResponseCode` is `success` then the Private Key used for Signing
4568 will have changed. If `updateDeviceCertResponseCode` is other than `success`, the
4569 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 ::=
    -- 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
    Certificate

  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	

4601 Table 13.5.7a: @UpdateDeviceCertificateonDevice.CommandPayload population

4602 @UpdateDeviceCertificateonDevice.ResponsePayload shall have the structure defined in Section 13.5.6, and the Device
 4603 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	

4604 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
SEC User Interface Services Schedule (Service Request) Reference	6.24
Valid Target Device(s)	ESME / GSME / GPF / CHF
Valid Business Originator role(s) for Command invocation (and so, for DLMS COSEM Commands, which Application Association is to be used for delivery of the Command to the 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:

- 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

- 4626 ○ those held in the {supplier, digitalSignature, management} Trust
 4627 Anchor Cell, if the target Device's deviceType does not equal
 4628 communicationsHubCommunicationsHubFunction;
- 4629 2. validate that keyUsage in CommandPayload is either digitalSignature only or
 4630 keyAgreement only. If this validation fails then the Device shall set
 4631 provideDeviceCertResponseCode to invalidKeyUsage, and process from step
 4632 5;
- 4633 3. confirm that the Device holds a certificate which (1) is for the keyUsage identified at
 4634 step 2, (2) contains in hwSerialNum a value equal to the Device's Entity Identifier and
 4635 (3) contains in subjectPublicKey the bit string representation of the Public Key
 4636 corresponding to the Current Private Key for this keyUsage. If this validation fails then
 4637 the Device shall set provideDeviceCertResponseCode to noCertificateHeld,
 4638 and process from step 5;
- 4639 4. retrieve the certificate identified in step 3. If this step fails then the Device shall set
 4640 provideDeviceCertResponseCode to certificateRetrievalFailure, and
 4641 process from step 5;
- 4642 5. create a Response according to the requirements of Section 13.6.7, apply the Response
 4643 Cryptographic Protection required for a Response of Message Category SME.C.C, and
 4644 send the Response.

4645 13.6.5 Response Processing

4646 Response Recipient Verification may be undertaken as specified in this GBCS for a
 4647 Response of the Message Category SME.C.C. The provideDeviceCertResponseCode
 4648 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
    }

```

```
4690
4691 ProvideDeviceCertResponseCode ::=
4692 {
4693     invalidKeyUsage (1),
4694     noCertificateHeld (2),
4695     certificateRetrievalFailure (3)
4696 }
4697
4698
4699 END
```

4700 **13.6.7 Constructing the @ProvideDeviceCertificateFromDevice.CommandPayload and**
4701 **@ProvideDeviceCertificateFromDevice.ResponsePayload**

4702 @ProvideDeviceCertificateFromDevice.CommandPayload shall have the structure defined in Section 13.6.6 and the Remote Party
4703 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

4704 Table 13.6.7a: @ProvideDeviceCertificateFromDevice.CommandPayload population

4705 @ProvideDeviceCertificateFromDevice.ResponsePayload shall have the structure defined in Section 13.6.6, and the Device
4706 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

4707 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'³². 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, GSME, GPF or Type 1 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

³² 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

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.

4778 Only one Device in a Method B Join is remotely instructed. Thus, the HAN exchanges
 4779 required by a Method B Join shall always be instigated by the Device receiving such a
 4780 Command. From Table 13.7.1.2, this is always a GSME or ESME except where a GPF is to
 4781 Join to a PPMID, IHD or CAD. Thus, the sequence of a Method B Join is that the ESME /
 4782 GSME / GPF:

- 4783 • is sent a Join Command containing the Entity Identifier of the Device to which it is to
 4784 Join and that other Device's `DeviceType`;
- 4785 • verifies the cryptographic protection on the Command and checks to make sure it is
 4786 well formed and valid;
- 4787 • updates its Device Log to include details of the new Device;
- 4788 • for an ESME³³, undertakes the key establishment process with the specified Device,
 4789 as per the ZSE specification. The constraint that Key Establishment has to involve
 4790 the ZSE Trust Center shall not be applied by Devices; and
- 4791 • creates and sends a Response detailing the success or otherwise of its actions.

4792 A Method C join does not require exchange of Messages between the two Devices for the
 4793 establishment of the shared secret. Thus the sequence of a Method C Join is that each of
 4794 the GSME and PPMID:

- 4795 • is sent a Join Command containing the Entity Identifier of the Device to which it is to
 4796 Join, that other Device's `DeviceType` and Key Agreement `Certificate`;
- 4797 • verifies the cryptographic protection on the Command and does checks to make sure
 4798 it is well formed and valid;
- 4799 • updates its Device Log to include details of the new Device;
- 4800 • checks there is a well-formed Device Certificate in the Command;
- 4801 • optionally calculates the shared secret using the Device Certificate of the other
 4802 Device (which is provided in the Command); and
- 4803 • creates and sends a Response detailing the success or otherwise of its actions.

4804 **13.7.1.3 The format of Message Payloads – informative**

4805 In common with other GBCS Remote Messages related to the management of Security
 4806 Credentials, the payloads of Commands and Responses defined in this Section 13.7.1.3 are
 4807 specified using ASN.1, with DER encoding to be applied to Command and Response
 4808 payloads.

4809 **13.7.2 Device Requirements**

4810 All Devices shall:

- 4811 • support the ZSE Key Establishment Cluster as specified in Annex C of the ZSE
 4812 cluster;
- 4813 • support 'Crypto Suite 2' as defined in the ZSE specification; and
- 4814 • use 'Crypto Suite 2' when undertaking any associated Key Establishment process.

4815 Devices shall not apply any restrictions on the types of Devices used in any associated Key
 4816 Establishment process, except for those specified in the GBCS. Specifically, the ZSE
 4817 constraint requiring Trust Center involvement shall not be applied (where 'Trust Center' has
 4818 the meaning defined in ZSE).

³³ The shared secret between the Communications Hub and the Type 2 Device / GSME established when the Device joined the HAN shall be used by the GPF to authenticate with the Device.

4819 An ESME shall be configured to be a ZSE 'Router', as defined in ZSE so that
 4820 communications between the ESME and Devices Joined to the ESME are not reliant on
 4821 availability of the Communications Hub.

4822 Pursuant to the requirements in the SMETS and the CHTS requirement, Devices shall only
 4823 communicate at an application layer with other Devices that are currently in their Device Log
 4824 and are permitted by Device Based Access Controls (DBAC) as defined at Section 13.7.3.
 4825 Such communications shall always be secured using the shared secrets established
 4826 pursuant to Sections 13.7.4.

4827 Application layer communications within the scope of the DBAC requirement are HAN Only
 4828 Messages, including provision of information to a PPMID or Type 2 Device. Note that HAN
 4829 Only Messages between a PPMID and GSME have a structure that is specified in this GBCS
 4830 in the corresponding Use Cases, and those relate only to Add Credit and Activate
 4831 Emergency Credit Commands and the corresponding Responses.

4832 Each entry in a non CHF Device Log shall contain the Entity Identifier of the Authorised
 4833 Device and its `deviceType`.

4834 The Entity Identifier of a Device with `DeviceType` of
 4835 `communicationsHubGasProxyFunction` shall be the EUI 64-bit identifier of the ZigBee
 4836 radio interface installed in the Communications Hub.

4837 13.7.3 Device Based Access Control

4838 In relation to information and functionality within SMETS, a Device shall, when it is a
 4839 recipient of a Command or request for information from another Device on its SMHAN, only
 4840 attempt to action that Command when:

- 4841 • the sending Device's Entity Identifier is in the recipient Device's Device Log;
- 4842 • the ZSE cryptographic protection on the Message is authenticated using the Shared
 4843 Secret / Shared Secret Key established with the sending Device; and
- 4844 • the Command or request for information is explicitly allowed by a cell in Tables
 4845 13.7.3a and 13.7.3b, in terms of the `DeviceType` of the sending (client) and
 4846 receiving (service) Device. The receiving Device shall determine the sending
 4847 Device's `DeviceType` by reference to its Device Log entry for that sending Device.

4848 Where a Device is a recipient of a Command or request for information from another Device
 4849 on its SMHAN that does not meet the access requirements of this Section 13.7.3, it shall:

- 4850 • generate an entry in the Security Log recording failed Authentication;
- 4851 • discard the Command or request for information without execution and without
 4852 sending a Response; and
- 4853 • send an Alert notifying the failed Authentication, constructed as specified in Section
 4854 6.2.4.2, populated with the relevant Alert Code from Section 16, to the Known
 4855 Remote Party specified in Table 16.2.

4856 An ESME shall not action any ZSE Local Change Supply command from a PPMID where
 4857 Proposed Supply Status has any value other than 0x02 ('Supply ON').

4858 For SMETS section 7.5.4.1 and 7.5.4.2 interactions, the PPMID and GSME shall comply
 4859 with Use Cases PCS01 and PCS02 respectively.

Device Name	Server / recipient	ESME	GSME	Comms Hub (GPF)	HCALCS	PPMID	Type 2 (IHD or CAD)
Client / sender	<code>deviceType</code>	1	0	3	4	5	6

ESME	1	-	-	-	5.6.4.1	-	-
GSME	0	-	-	-	-	-	-
Comms Hub (GPF)	3	-	Request for Information	-	-	-	-
HCALCS	4	8.5.2.1	-	-	-	-	-
PPMID	5	7.5.5.1 7.5.5.2 7.5.5.3 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	Control HAN Connected Auxiliary Load Control Switch	Load Control Event
8.5.2.1	Request Control of a HAN Connected Auxiliary Load Control Switch	Get Scheduled Events
7.5.5.1	Request Emergency Credit Activation	Select Available Emergency Credit
7.5.5.2	Request to Add Credit	Consumer Top Up
7.5.5.3	Request to Enable ESME Supply	Local Change Supply
7.5.4.1	Request Emergency Credit Activation	Select Available Emergency Credit
7.5.4.2	Request to Add Credit	Consumer Top Up

Table 13.7.3b: Mapping of Table 13.7.3a command references to SMETS names and ZSE

13.7.4 Use Case Requirements

This Section 13.7.4 details requirements which shall be complied with for all Join or Unjoin related Use Cases.

13.7.4.1 Use Cases covered

The types of Join Device related Messages, the Grouping names used in this Section 13.7.4, the associated Message Category and the valid recipient *deviceType* for each shall be as specified in Table 13.7.4.1³⁴. The SEC User Interface Services Schedule (Service Request) Reference for all Join Use Cases shall be 8.7, and for Unjoin Use Cases shall be 8.8.

Message Code	Use Case Name	Valid recipient <i>deviceType</i>	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	type1HANConnectedAuxiliaryLoadControlSwitch type1PrepaymentInterfaceDevice	Join Device	SME.C.C	Access Control Broker where the

³⁴ To avoid duplication of specification, the Use Cases here are grouped together, and the standard Use Case cross reference table is not used.

	(non Meter)				Command is addressed to a PPMID; Supplier otherwise
0x000E	CS03B Method B Join	gSME eSME communicationsHubGasProxyFunction	Join Device	SME.C.N C	Supplier, Access Control Broker
0x00AF	CS03C Method C Join	gSME type1PrepaymentInterfaceDevice	Join Device	SME.C.C	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 communicationsHubGasProxyFunction	Unjoin Device	SME.C.N C	Supplier, Access Control Broker
0x0013	CS07 Read Device Join Details	gSME eSME communicationsHubGasProxyFunction type1HANConnectedAuxiliaryLoadControlSwitch type1PrepaymentInterfaceDevice		SME.C.N C	Supplier, Access Control Broker

4871 Table 13.7.4.1: Join Device related Commands, Grouping and Message Categories

4872 **13.7.4.2 Join Device Command and Response Processing**

4873 **13.7.4.2.1 Construction of Commands**

4874 ‘Join Device’ Command Payloads shall be constructed as specified in Section 13.7.4.5.2 and
 4875 Cryptographic Protection I and Cryptographic Protection II shall be applied as required for a
 4876 Command of the relevant Message Category.

4877 For a Command (1) which complies with either Use Case ‘CS03A2 Method A Join (non
 4878 Meter)’ or Use Case ‘CS03C Method C Join’ and (2) where the Device to which it is
 4879 addressed has a deviceType equal to type1PrepaymentInterfaceDevice, the
 4880 Access Control Broker’s Digital Signing Private Key shall be used in generating the KRP
 4881 Signature.

4882 **13.7.4.2.2 Device processing of Commands and Response handling**

4883 The Device receiving a ‘Join Device’ Command shall undertake processing steps in the
 4884 sequence defined in this Section 13.7.4.2.2. Should a step after step 1 be unsuccessful, the
 4885 Device shall create a Response according to the requirements of Section 13.4.7, apply the
 4886 Response Cryptographic Protection required for a Response of the relevant Message
 4887 Category, and send the Response and shall not undertake any further steps defined in this
 4888 Section 13.7.4.2.2.

4889 In processing a ‘Join Device’ Command, the Device shall:

- 4890 1. undertake Command Authenticity and Integrity Verification as required for a Command
- 4891 of this Message Category. The Security Credentials used to verify Cryptographic
- 4892 Protection 1 shall be:

- 4893 o those held in the {accessControlBroker, digitalSignature,
4894 management} Trust Anchor Cell, if deviceType equals
4895 type1PrepaymentInterfaceDevice; or
- 4896 o those held in the {supplier, digitalSignature, management} Trust
4897 Anchor Cell, if deviceType does not equal
4898 type1PrepaymentInterfaceDevice;
- 4899 2. verify the joinMethodAndRole as specified in Section 13.7.4.5.3;
- 4900 3. add the otherDeviceEntityIdentifier and otherDeviceType to its Device Log
4901 as specified in Section 13.7.4.5.4;
- 4902 4. if deviceType is eSME then undertake Key Establishment with the other Device as
4903 specified in Section 13.7.4.5.5;
- 4904 5. if joinMethodAndRole is methodC, and so the join is between a gSME and a
4905 type1PrepaymentInterfaceDevice, check that otherDeviceCertificate is
4906 present and validly structured. If the check succeeds the Device shall store, linked to
4907 this Device Log entry, details relating to otherDeviceCertificate, such that the
4908 Device is able to use subsequently the Shared Secret derived from
4909 otherDeviceCertificate and its own Private Key Agreement Key. If this check
4910 fails the Device shall set joinResponseCode to invalidOrMissingCertificate
4911 and processing shall be unsuccessful;
- 4912 6. set joinResponseCode to success, create a Response according to the
4913 requirements of Section 13.4.7, apply the Response Cryptographic Protection required
4914 for a Response of the relevant Message Category, and send the Response.

4915 13.7.4.2.3 Response Processing

4916 Response Recipient Verification may be undertaken as specified in this GBCS for a
4917 Response of the relevant Message Category. The joinResponseCode field in the
4918 Response may be decoded according to the ASN.1 definitions at Section 13.7.4.5.1.

4919 13.7.4.3 'Unjoin Device' Command and Response Processing

4920 13.7.4.3.1 Construction of Commands

4921 'Unjoin Device' Command Payloads shall be constructed as specified in Section 13.7.4.6.2
4922 and Cryptographic Protection I and Cryptographic Protection II shall be applied as required
4923 for a Command of the relevant Message Category.

4924 For a Command where the Device to which it is addressed has a deviceType equal to
4925 type1PrepaymentInterfaceDevice, the Access Control Broker's Digital Signing Private
4926 Key shall be used in generating the KRP Signature.

4927 13.7.4.3.2 Device processing of Commands and Response handling

4928 The Device receiving an 'Unjoin Device' Command shall undertake processing steps in the
4929 sequence defined in this Section 13.7.4.3.2.

4930 In processing an 'Unjoin Device' Command, the Device shall:

- 4931 1. undertake Command Authenticity and Integrity Verification as required for a Command
4932 of this Message Category. The Security Credentials used to verify Cryptographic
4933 Protection I shall be:
 - 4934 o those held in the {accessControlBroker, digitalSignature,
4935 management} Trust Anchor Cell, if deviceType equals
4936 type1PrepaymentInterfaceDevice; or

- 4937 o those held in the {supplier, digitalSignature, management} Trust
- 4938 Anchor Cell, if deviceType does not equal
- 4939 type1PrepaymentInterfaceDevice;
- 4940 2. set unjoinResponseCode to success;
- 4941 3. verify the otherDeviceEntityIdentifier matches an Entity Identifier currently
- 4942 recorded in its Device Log. If it does not then set unjoinResponseCode to
- 4943 otherDeviceNotInDeviceLog and process from step 5; otherwise process from step
- 4944 4;
- 4945 4. delete all information from the entry in its Device Log that has the same Entity Identifier
- 4946 as otherDeviceEntityIdentifier along with all shared cryptographic material
- 4947 related to that entry. If the deletion does not succeed, set unjoinResponseCode to
- 4948 otherFailure; and
- 4949 5. Create a Response according to the requirements of Section 13.7.4.6.2, apply the
- 4950 Response Cryptographic Protection required for a Response of the relevant Message
- 4951 Category, and send the Response.

4952 13.7.4.3.3 Response Processing

4953 Response Recipient Verification may be undertaken as specified in this GBCS for a

4954 Response of the relevant Message Category. The unjoinResponseCode field in the

4955 Response may be decoded according to the ASN.1 definitions at Section 13.7.4.6.1.

4956 13.7.4.4 'CS07 Read Device Join Details' Command and Response Processing

4957 13.7.4.4.1 Construction of Commands

4958 'CS07 Read Device Join Details' Command Payloads shall be constructed as specified in

4959 Section 13.7.4.7 and Cryptographic Protection II shall be applied as required for a Command

4960 of the SME.C.NC Message Category.

4961 13.7.4.4.2 Device processing of Commands and Response handling

4962 The Device receiving a 'CS07 Read Device Join Details' Command shall undertake

4963 processing steps in the sequence defined in this Section 13.7.4.4.2.

4964 In processing a 'CS07 Read Device Join Details' Command, the Device shall:

- 4965 1. undertake Command Authenticity and Integrity Verification as required for a Command
- 4966 of the SME.C.NC Message Category;
- 4967 2. set readLogResponseCode to success;
- 4968 3. attempt to read the Entity Identifier and deviceType for each of the entries in its
- 4969 Device Log. If the reading does not succeed for all entries, set readLogResponseCode
- 4970 to readFailure; otherwise populate deviceLogEntries using the data read from its
- 4971 Device Log; and
- 4972 4. create a Response according to the requirements of Section 13.7.4.7, apply the
- 4973 Response Cryptographic Protection required for a Response of the SME.C.NC Message
- 4974 Category, and send the Response.

4975 13.7.4.4.3 Response Processing

4976 Response Recipient Verification may be undertaken as specified in this GBCS for a

4977 Response of the SME.C.NC Message Category. The readLogResponseCode and

4978 deviceLogEntries fields in the Response may be decoded according to the ASN.1

4979 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 ::=
{
    -- 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 didn't,
-- 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),

```

```
5045         invalidMessageCodeForJoinMethodAndRole      (1),
5046         invalidJoinMethodAndRole                      (2),
5047         incompatibleWithExistingEntry                 (3),
5048         deviceLogFull                                 (4),
5049         writeFailure                                  (5),
5050         keyAgreementNoResources                       (6),
5051         keyAgreementUnknownIssuer                    (7),
5052         keyAgreementUnsupportedSuite                  (8),
5053         keyAgreementBadMessage                       (9),
5054         keyAgreementBadKeyConfirm                    (10),
5055         invalidOrMissingCertificate                   (11)
5056     }
5057
5058     END
```


5059 *13.7.4.5.2 Constructing the @JoinDevice.CommandPayload and of @JoinDevice.ResponsePayload*

5060 @JoinDevice.CommandPayload shall have the structure defined in Section 13.7.4.5.1, and the Remote Party constructing the Command
5061 shall populate with values according to Table 13.7.4.5.2a.

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
@JoinDevice.CommandPayload	SEQUENCE			
joinMethodAndRole	INTEGER	methodAInitiator (0), methodAResponder (1), methodB (2), methodC (3)		See Section 13.7.4.5.3 for valid values
otherDeviceEntityIdentifier	OCTET STRING	Entity Identifier	Mandatory	The Entity Identifier of the Device which is to be entered in this Device's Device Log
otherDeviceType	INTEGER	gSME (0), eSME (1), communicationsHubCommunicationsHubFunction (2), communicationsHubGasProxyFunction (3), type1HANConnectedAuxiliaryLoadControlSwitch (4), type1PrepaymentInterfaceDevice (5), type2 (6)	Mandatory	The DeviceType of the Device which is to be entered in this Device's Device Log
otherDeviceCertificate	Certificate	The Key Agreement Certificate currently in use by the other Device.	OPTIONAL	The other Device's Key Agreement certificate, which shall only be present if and only if this is a join between a gSME and a type1PrepaymentInterfaceDevice. Certificate shall be as defined in IETF RFC 5912.

5062 Table 13.7.4.5.2a: @JoinDevice.CommandPayload population

5063 @JoinDevice.ResponsePayload shall have the structure defined in Section 13.7.4.5.1, and the Remote Party constructing the Command
5064 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	

5065 Table 13.7.4.5.2b: @JoinDevice.ResponsePayload population

5066 *13.7.4.5.3 Verification of joinMethodAndRole*

5067 The Device shall first verify the joinMethodAndRole specified in the Command Payload against the Message Code specified in the Grouping
5068 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
5069 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

5070 Table 13.7.4.5.3a: Valid deviceMethod and joinMethodAndRole against Message Code

5071 The Device receiving a Join Device Command shall verify joinMethodAndRole against its own DeviceMethod and the DeviceType
5072 specified in the otherDeviceType parameter of the Command according to the requirements of the remainder of this Section 13.7.4.5.3.

5073 If joinMethodAndRole is methodB then the Device's verification of joinMethodAndRole shall be successful if there is a cell identified by
5074 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
5075 cell contains 'success'. Otherwise, the verification shall fail and JoinResponseCode in the Response shall be set to the value
5076 invalidJoinMethodAndRole.

	otherDeviceType		
	communicationsHub GasProxyFunction	type1PrepaymentInterfaceDevice	type2
DeviceType of Device to which the Command is addressed			
gSME	Success	-	-
eSME	-	-	Success
communicationsHubGasProxyFunction	-	Success	Success

5077 Table 13.7.4.5.3b: joinMethodAndRole is methodB

5078 If joinMethodAndRole is methodAInitiator then the Device's verification of joinMethodAndRole shall be successful if there is a cell
 5079 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,
 5080 and that cell contains 'success'. Otherwise, the verification shall fail and JoinResponseCode in the Response shall be set to the value
 5081 invalidJoinMethodAndRole.

	otherDeviceType	
	type1HANConnected AuxiliaryLoadControlSwitch	type1Prepayment InterfaceDevice
DeviceType of Device to which the Command is addressed		
eSME	Success	Success

5082 Table 13.7.4.5.3c: joinMethodAndRole is methodB

5083 If joinMethodAndRole is methodAResponder then the Device's verification of joinMethodAndRole shall be successful if there is a cell
 5084 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,
 5085 and that cell contains 'success'. Otherwise, the verification shall fail and JoinResponseCode in the Response shall be set to the value
 5086 invalidJoinMethodAndRole.

otherDeviceType	
	eSME
DeviceType of Device to which the Command is addressed	
type1HANConnectedAuxiliaryLoadControlSwitch	Success
type1PrepaymentInterfaceDevice	Success

5087 Table 13.7.4.5.3d: joinMethodAndRole is methodAResponder

5088 If joinMethodAndRole is methodC then the Device's verification of joinMethodAndRole shall be successful if there is a cell identified by
 5089 its own DeviceType and the value of otherDeviceType (as identified in the Command) in Table 13.7.4.5.3e and that cell contains
 5090 'success'. Otherwise, the verification shall fail and JoinResponseCode in the Response shall be set to the value
 5091 invalidJoinMethodAndRole.

otherDeviceType		
	type1PrepaymentInterfaceDevice	gSME
DeviceType of Device to which the Command is addressed		
gSME	Success	-
type1PrepaymentInterfaceDevice	-	Success

5092 Table 13.7.4.5.3e: joinMethodAndRole is methodB

5093 *13.7.4.5.4 Adding the otherDeviceEntityIdentifier and otherDeviceType to the Device Log*

5094 The Device shall undertake the following steps in the sequence specified:

- 5095 1. if the otherDeviceEntityIdentifier matches an Entity Identifier currently recorded in its Device Log, then the Device shall compare
 5096 deviceType in that log entry with otherDeviceType. If the Device types match then the addition is successful and processing within
 5097 this Section 13.7.4.5.4 shall cease; otherwise the Device shall set joinResponseCode to incompatibleWithExistingEntry and
 5098 processing within this Section 13.7.4.5.4 shall cease;
- 5099 2. the Device shall check if there is capacity for an additional entry in its Device Log. If there is not, the Device shall set joinResponseCode
 5100 to deviceLogFull and processing within this Section 13.7.4.5.4 shall cease; and

5101 3. the Device shall attempt to create a new Device Log entry using `otherDeviceEntityIdentifier` and `otherDeviceType`. If that
 5102 entry is not successfully created, the Device shall set `joinResponseCode` to `writeFailure`.

5103 *13.7.4.5.5 Undertaking Key Establishment with the other Device*

5104 The Device shall initiate, and attempt to complete, Key Establishment according to the ZSE requirements. The initiating Device shall wait a
 5105 minimum of two seconds before timing out any key establishment operation.

5106 Should there be errors that result in that process not completing, the Device shall set `joinResponseCode` to the value specified by Table
 5107 13.7.4.5.5.

ZSE Response Code ³⁵	Value of <code>joinResponseCode</code>
NO_RESOURCES	<code>keyAgreementNoResources</code>
UNKNOWN_ISSUER	<code>keyAgreementUnknownIssuer</code>
UNSUPPORTED_SUITE	<code>keyAgreementUnsupportedSuite</code>
BAD_MESSAGE	<code>keyAgreementBadMessage</code>
BAD_KEY_CONFIRM	<code>keyAgreementBadKeyConfirm</code>

5108 Table 13.7.4.5.5: `joinResponseCode` mapping to ZSE Responses

5109 *13.7.4.6 Component Requirements – Unjoin*

5110 *13.7.4.6.1 Unjoin Command and Response payloads – structure definition*

5111 Each instance of `@UnjoinDevice.CommandPayload` and of `@UnjoinDevice.ResponsePayload` shall be an octet string containing the
 5112 DER encoding of the populated structure defined in this Section 13.7.4.6.1 which specifies the structure in ASN.1 notation.

```

5113 UnjoinDevice DEFINITIONS ::= BEGIN
5114
5115 CommandPayload ::=
5116     -- specify the Entity Identifier of the Device for which authorisation
5117     -- is to be removed
5118
5119     OtherDeviceEntityIdentifier ::= OCTET STRING
5120
5121 ResponsePayload ::=

```

³⁵ As defined in the ZSE specification

```

5122
5123      -- detail whether the Command successful executed or, if it didn't,
5124      -- what the failure reason was
5125
5126 UnjoinResponseCode ::=
5127 {
5128     success                      (0),
5129     otherDeviceNotInDeviceLog    (1),
5130     otherFailure                 (2)
5131 }
5132
5133 END

```

5134 13.7.4.6.2 Constructing the @UnjoinDevice.CommandPayload and of @UnjoinDevice.ResponsePayload

5135 @UnjoinDevice.CommandPayload shall have the structure defined in Section 13.7.4.6.1, and the Remote Party constructing the Command
 5136 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

5137 Table 13.7.4.6.2a: @UnjoinDevice.CommandPayload population

5138 @UnjoinDevice.ResponsePayload shall have the structure defined in Section 13.7.4.6.1, and the Remote Party constructing the
 5139 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),	Mandatory	Shall be populated according to the processing defined in Section 13.7.4.3

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
		otherDeviceNotInDeviceLog (1), (1), otherFailure (2) (2)		

5140 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
  {
    deviceIdentifier          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

```

5187

5188 **13.8 GCS59 / 62 GPF Device Log Backup and Restore**5189 **13.8.1 Introduction to GPF Device Log Backup and Restore –**
5190 **informative**5191 **13.8.1.1 The role of pair-wise authorisation – informative**

5192 This Section 13.8 includes the Use Cases related to the backing up and restoring of the
5193 GPF's Device Log. This is to cater for situation where the existing Communications Hub
5194 fails and has to be replaced.

5195 In summary:

- 5196 • a GPF sends an Alert whenever its Device Log changes (unless the change is as a
5197 result of a restore of the Device Log). That Alert contains the contents of the GPF's
5198 Device Log after the change has been made; and
- 5199 • the Restore GPF Device Log Command shall contain the same structure of Device
5200 Log contents. If successful, the Command will place those contents in to the GPF's
5201 Device Log and will have triggered the processing required to authorise the specified
5202 Devices application layer interaction with the GPF, where required.

5203 **13.8.1.2 The format of Message Payloads – informative**

5204 In common with other GBCS Remote Messages related to the management of Security
5205 Credentials, the Payloads of Alerts, Commands and Responses defined in this Section 13.8
5206 are specified using ASN.1, with DER encoding to be applied to Command and Response
5207 payloads.

5208 Each entry in a GPF Device Log shall contain the Entity Identifier of the Authorised Device
5209 and its `deviceType`.

5210 **13.8.2 GCS62 Backup GPF Device Log**5211 **13.8.2.1 Description**

5212 This Section 13.8.2 covers the creation, validation and processing of Alerts resulting from
5213 changes to the GPF Device Log. One such Alert shall be generated each time that the GPF
5214 Device Log changes, except where the change arises from a GPF Device Log Restore
5215 Command.

5216 **13.8.2.2 Use Case Cross References**

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Alert
Message Type Category	SME.A.NC
Capable of future dated invocation?	N/A
Protection Against Replay Required?	N/A
SEC User Interface Services Schedule (Service Request) Reference	8.12
Valid Initiating Device(s)	GPF
Valid Business Target role(s) for Alert	Access Control Broker
Valid Response Recipient role(s) (only for Messages Authorised by the Access Control	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

5217 Table 13.8.2.2: Use Case Cross References for GPF Device Log Backup Alert

5218 **13.8.2.3 Construction of Alerts**

5219 GPF Device Log Backup Alert Payloads shall be constructed according to the requirements
5220 of Section 13.8.4.1 and populated as specified in Table 13.8.2.3.

5221 MAC Header, Grouping Header and SMD-KRP MAC shall be populated as required for an
5222 Alert of the SME.A.NC Message Category, with the Message Code being 0x00B2. Note that
5223 the Business Target ID in the Grouping Header shall always contain the Entity Identifier of
5224 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

5225 Table 13.8.2.3: @GPFDeviceLog.BackupAlertPayload population

5226 **13.8.2.4 Processing of Alerts**

5227 SMD-KRP MAC may be verified by the Access Control Broker as per Section 6.8.3.

5228 **13.8.3 GCS59 GPF Device Log Restore**

5229 **13.8.3.1 Description**

5230 This section covers the creation, validation and processing of Commands to restore the GPF Device Log, and the creation and validation of the
5231 corresponding Response.

5232 **13.8.3.2 Use Case Cross References**

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Command and Response

Cross Reference	Value
Message Type Category	SME.C.NC
Capable of future dated invocation?	No
Protection Against Replay Required?	Yes
SEC User Interface Services Schedule (Service Request) Reference	8.12
Valid Target Device(s)	GPF
Valid Business Originator role(s) for Command	Access Control Broker
Valid Response Recipient role(s) (only for Messages Authorised by the Access Control Broker on behalf of parties not known to the Device) [Remote Party Messages Only]	N/A
Valid initiating Device type(s) [HAN Only Messages]	N/A
Protocol	ASN.1

5233 Table 13.8.3.2: Use Case Cross References for GPF Device Log Restore

5234 **13.8.3.3 Construction of Command**

5235 GPF Device Log Restore Command Payloads shall be constructed according to the requirements of Section 13.8.4.1 and populated as
5236 specified in Table 13.8.3.3.

5237 MAC Header, Grouping Header, KRP Signature and ACB-SMD MAC shall be populated as required for a Command of the SME.C.C Message
5238 Category.

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
@GPFDeviceLog.RestoreComm andPayload	SEQUENCE			
deviceLogEntries	SEQUENCE OF		OPTIONAL	There may be 0, 1 or many entries in the Log. The following two fields will be repeated as many times as there are Device Log Entries. Note that there would be no effect if the Command had no deviceLogEntries.

Attribute name	Data Type	Value (blank cells mean the command specific value is derived by the encoding process)	Mandatory, OPTIONAL or DEFAULT value	Notes
deviceEntityIdentifier	OCTET STRING	Entity Identifier	Mandatory as part of each entry that is present	The Entity Identifier of the Device to which this entry relates.
deviceType	INTEGER	type1PrepaymentInterfaceDevice (5), type2 (6)	Mandatory as part of each entry that is present	The DeviceType of the Device to which this entry relates. These are the only valid entries for the GPF Device Log. Note that the GSME does not need to be in the GPF's Device Log, since the GPF only receives information from the GSME.

5239 Table 13.8.3.3: @GPFDeviceLog.RestoreCommandPayload population

5240 **13.8.3.4 Device processing of Command and Response handling**

5241 The GPF receiving a GPF Device Log Restore Command shall undertake processing steps in the sequence defined in this Section 13.8.3.4.
5242 The Device shall undertake Command Authenticity and Integrity Verification as required for a Command of this Message Category, and then, if
5243 successful, for each DeviceLogEntry in deviceLogEntries, shall:

- 5244 1. set deviceLogEntry in the corresponding ResponseOutcome to the values of this DeviceLogEntry in deviceLogEntries;
- 5245 2. set joinResponseCode in the corresponding ResponseOutcome to success;
- 5246 3. if the deviceEntityIdentifier matches an Entity Identifier currently recorded in its Device Log, compare deviceType in that log
5247 entry with otherDeviceType. If the Device types match then the addition is successful and processing of this DeviceLogEntry shall
5248 cease; otherwise the Device shall set joinResponseCode to incompatibleWithExistingEntry and processing of this
5249 DeviceLogEntry shall cease;
- 5250 4. check if there is capacity for an additional entry in its Device Log. If there is not, the Device shall set joinResponseCode to
5251 deviceLogFull and processing of this DeviceLogEntry shall cease; and
- 5252 5. attempt to create a new Device Log entry using deviceEntityIdentifier and deviceType. If that entry is not successfully created,
5253 the Device shall set joinResponseCode to writeFailure and processing of this DeviceLogEntry shall cease.

5254 Once all DeviceLogEntry in deviceLogEntries have been processed, the GPF shall populate the Response Payload according to the
5255 requirements of Section 13.8.4.1 using the ResponseOutcomes produced by the processing in this Section 13.8.3.4, construct MAC Header,
5256 Grouping Header and apply the Response Cryptographic Protection required for a Response of the SME.C.NC Message Category, and send
5257 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

```



```

5298 {
5299     -- for each DeviceLog Entry, detail whether the Command successfully executed or, if it didn't, what the failure reason was
5300
5301     restoreOutcomes                                SEQUENCE OF RestoreOutcome
5302 }
5303
5304 RestoreOutcome ::=                                SEQUENCE
5305 {
5306     deviceLogEntry                                DeviceLogEntry,
5307     joinResponseCode                              JoinResponseCode
5308 }
5309
5310 DeviceType ::=                                    INTEGER
5311 {
5312     gSME                                           (0),
5313     eSME                                           (1),
5314     communicationsHubCommunicationsHubFunction    (2),
5315     communicationsHubGasProxyFunction             (3),
5316     type1HANConnectedAuxiliaryLoadControlSwitch   (4),
5317     type1PrepaymentInterfaceDevice                (5),
5318     type2                                          (6)
5319 }
5320
5321 JoinResponseCode ::=                              INTEGER
5322 {
5323     success                                         (0),
5324     invalidMessageCodeForJoinMethodAndRole         (1),
5325     invalidJoinMethodAndRole                      (2),
5326     incompatibleWithExistingEntry                 (3),
5327     deviceLogFull                                 (4),
5328     writeFailure                                  (5),
5329     keyAgreementNoResources                       (6),
5330     keyAgreementUnknownIssuer                    (7),
5331     keyAgreementUnsupportedSuite                  (8),
5332     keyAgreementBadMessage                       (9),
5333     keyAgreementBadKeyConfirm                    (10),
5334     invalidOrMissingCertificate                   (11)
5335 }
5336 END

```

14 Apply Prepayment Top Up to an ESME or GSME

14.1 Defined Terms

The following terms used in this Section 14 shall have the meanings defined in this Table 14.1.

Defined Term	Meaning
Currency Unit	Shall be either GB Pound or European Central Bank Euro
Maximum Credit Threshold	Shall be the maximum value of any single 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 20 th digit of the UTRN
UTRN Counter Cache	Shall be an array of 100 entries, each entry containing an unsigned integer of 32 bits in length and an associated flag to indicate whether the UTRN Counter represented by the integer relates to a locally entered Prepayment Top Up, a network delivered Prepayment Top Up or has been set as a floor value on execution of an Update Security Credentials Command The array shall be arranged as a circular buffer such that, when full, further writes shall cause the lowest numerical value entry to be overwritten
UTRN Counter	The 32 most significant bits of the Originator Counter

Table 14.1: Meanings of Defined Terms

14.2 Description – informative

This section covers the application of a Prepayment Top Up, that has been purchased for a particular ESME or GSME, to that ESME or GSME.

It covers four options:

- applying a Prepayment Top Up to an ESME without consumer intervention;
- applying a Prepayment Top Up to a GSME without consumer intervention;
- applying a Prepayment Top Up to an ESME or GSME with consumer entry of a numeric code on the ESME or GSME; and
- applying a Prepayment Top Up to an ESME or GSME with consumer entry of a numeric code on a PPMID.

5353 Some requirements are common to all four options. Accordingly, this Section 14 is split in to
5354 five subsections:

- 5355 • an initial subsection covering requirements common to all four options; and
- 5356 • four subsequent subsections covering one option in each subsection.

5357 By way of context:

- 5358 • any Prepayment Top Up Message is a Remote Party Command in GBCS terms
5359 (because it is from a Remote Party to a GSME or ESME). The means of delivery
5360 (typing in on meter, typing in on PPMID, sending over WAN, etc.) does not affect this
5361 classification;
- 5362 • as a Remote Party Command, it must result in the GSME or ESME generating a
5363 Response back to the Remote Party who issued it (so the Supplier), unless there is
5364 an Authentication failure (in which case the Supplier has to be sent an Alert), as per
5365 SMETS and CHTS;
- 5366 • because the ranges are exclusive, the Originator Counter in Prepayment Top Up
5367 transactions cannot collide with the Originator Counter in any other transaction; and
- 5368 • there is no requirement to include the Device's ID explicitly in the locally entered
5369 transaction, so a PPMID joined to more than one Smart Meter will need to allow the
5370 Consumer to pick which Smart Meter the Prepayment Top Up is for.

5371 14.3 Common Requirements

5372 14.3.1 Construction of the PPTD

5373 The PPTD shall be a 19 decimal digit integer. The most significant two digits of the PPTD
5374 shall always be between 73 and 96, which shall be constructed and represented according
5375 to the requirements of this Section 14.3.1.

5376 The decimal representation of the PPTD shall be the result of the addition of
5377 7,394,156,990,786,306,048 to the decimal representation of the PTUT.

5378 14.3.2 Construction of the PTUT

5379 The PTUT shall be an unsigned 64 bit integer (so 8 octets), which shall be constructed and
5380 represented according to the requirements of this Section 14.3.2.

5381 The bits within the PTUT shall be numbered from 63 for the most significant bit through to 0
5382 for the least significant bit.

5383 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 56-47 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 Value will be interpreted as GB Pounds

PTUT component	Value	Bits	Note
	0b01 if PTUT Value is to be interpreted as multiples of Currency Unit.		
PTUT Value	The quantum of the PTUT expressed as an unsigned binary number of 13 bits in length, so with leading binary zeros where required	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 Message Identifier 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 Message Identifier 32 most significant bits of the PTUT	

5404 Table 14.3.5: Validation of the PTUT Supplier MAC

5405 14.3.6 Checking the UTRN Counter against the UTRN Counter 5406 Cache

5407 The Device shall set the UTRN Counter to be the 32 most significant bits of the Originator
5408 Counter.

5409 The Device shall check that the UTRN Counter is strictly numerically greater than the
5410 numerically lowest value in the UTRN Counter Cache, and is not equal to any value in the
5411 UTRN Originator Counter Cache.

5412 14.3.7 Updating the UTRN Counter Cache

5413 Where the Prepayment Top Up is successfully applied and prior to sending any Response,
5414 the Device shall add a new entry to the UTRN Counter Cache whose UTRN Counter value
5415 shall be set to the 32 most significant bits of Originator Counter and whose flag shall be set
5416 to record this Prepayment Top Up either as a network delivered Prepayment Top Up or as a
5417 locally entered Prepayment Top Up, as appropriate.

5418 14.3.8 Validating the Maximum Credit Values

5419 14.3.8.1 Maximum Credit Threshold

5420 The Device shall ensure that the top-up value specified by PTUT Value Class and PTUT
5421 Value does not exceed the Device's Maximum Credit Threshold parameter.

5422 14.3.8.2 Maximum Meter Balance Threshold

5423 The Device shall ensure that the top-up value specified by PTUT Value Class and PTUT
5424 Value when added to the Device's Credit Balance does not exceed the Device's Maximum
5425 Meter Balance Threshold parameter.

5426 14.3.9 Validating the PTUT Sub-Class

5427 The Device shall ensure that the value specified by PTUT Sub-Class is of value 0b0000.

5428 14.4 CS01a Applying a Prepayment Top Up to an ESME 5429 without consumer intervention

5430 14.4.1 Description

5431 This section covers the application of a Prepayment Top Up that has been bought for an
5432 ESME to that ESME, in the case where the consumer does not enter any details on Devices
5433 in their premises.

5434 **14.4.2 Use Case Cross References**

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Command and Response
Message Type Category	SME.C.NC but with additional cryptographic processing specified in Sections 14.3.4 and 14.3.5
Capable of future dated invocation?	No
Protection Against Replay Required?	The Protection Against Replay mechanisms for Prepayment Top Ups are specified in Section 14.3.6. The Protection Against Replay mechanisms specified elsewhere in the GBCS do not apply
SEC User Interface Services Schedule (Service Request) Reference	2.2
Valid Target Device(s)	ESME
Valid Business Originator role(s) for Command invocation (and so, for DLMS COSEM Commands, which Application Association is to be used for delivery of the Command to the ESME) [Remote Party Messages Only]	Supplier
Valid Response Recipient role(s) (only for Messages Authorised by the Access Control Broker on behalf of parties not known to the Device) [Remote Party Messages Only]	N/A
Valid initiating Device type(s) [HAN Only Messages]	N/A
Protocol	DLMS COSEM

5435 Table 14.4.2: Use Case Cross References for Prepayment Top Up to an ESME without consumer
5436 intervention5437 **14.4.3 Pre-conditions**

5438 None.

5439 **14.4.4 Detailed Steps**5440 The Device shall undertake the checks set out in this Section 14.4.4 in the sequence laid
5441 out:

- 5442 • only once all checks in Section 6.2.4.1.1 have been successfully completed; and
- 5443 • before undertaking any other processing of the Command.

5444 If any of the checks specified in this Section 14.4.4 fail, the Device shall not carry out further
5445 checks, and the requirements of Section 6.2.4.2 shall apply. Otherwise, processing shall
5446 continue as per the requirements of Section 6.2.4.1.2. Where that check is successful,
5447 processing shall continue as below.

5448 **14.4.4.1 Verifying against the maximum credit values**

5449 The Device shall carry out the checks specified in Section 14.3.8.1 and Section 14.3.8.2.

5450 **14.4.4.2 Verifying the Originator Counter**

5451 The Device shall verify the Originator Counter against the UTRN Counter Cache according
5452 to Section 14.3.6.

5453 **14.4.4.3 Validating the PTUT Supplier MAC**

5454 The Device shall validate the PTUT Supplier MAC according to Section 14.3.5.

5455 **14.4.5 Response Construction**

5456 At the Response Construction stage, the Device shall first update the UTRN Counter Cache
5457 according to Section 14.3.7, and shall then populate the Response according to the
5458 requirements of the Message Template for CS01a.

5459 **14.5 CS01b Applying a Prepayment Top Up to a GSME**
5460 **without consumer intervention**

5461 **14.5.1 Description**

5462 This section covers the application of a Prepayment Top Up that has been bought for a
5463 GSME to that GSME, in the case where the consumer does not enter any details on Devices
5464 in their premises, except for the additional processing defined in this section.

5465 **14.5.2 Use Case Cross References**

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Command and Response
Message Type Category	See Table 14.4.2
Capable of future dated invocation?	No
Protection Against Replay Required?	See Table 14.4.2
SEC User Interface Services Schedule (Service Request) Reference	2.2
Valid Target Device(s)	GSME
Valid Business Originator role(s) for Command invocation (and so, for DLMS COSEM Commands, which Application Association is to be used for delivery of the Command to the ESME) [Remote Party Messages Only]	Supplier
Valid Response Recipient role(s) (only for Messages Authorised by the Access Control Broker on behalf of parties not known to the Device) [Remote Party Messages Only]	N/A
Valid initiating Device type(s) [HAN Only Messages]	N/A
Protocol	GBZ

5466 Table 14.5.2: Use Case Cross References for Prepayment Top Up to a GSME without consumer
5467 intervention

5468 **14.5.3 Pre-conditions**

5469 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 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. An Alert as specified in SMETS for locally entered commands is not required
Capable of future dated invocation?	No
Protection Against Replay Required?	See Table 14.4.2

Cross Reference	Value
SEC User Interface Services Schedule (Service Request) Reference	N/A
Valid Target Device(s)	ESME or GSME
Valid Business Originator role(s) for Command invocation (and so, for DLMS COSEM Commands, which Application Association is to be used for delivery of the Command to the Device) [Remote Party Messages Only]	Supplier
Valid Response Recipient role(s) (only for Messages Authorised by the Access Control Broker on behalf of parties not known to the Device) [Remote Party Messages Only]	N/A
Valid initiating Device type(s) [HAN Only Messages]	N/A
Protocol	Outside of protocols since entered via User Interface

5498 Table 14.6.2: Use Case Cross References for Prepayment Top Up through consumer UTRN entry

5499 14.6.3 Pre-conditions

5500 None.

5501 14.6.4 Detailed Steps

5502 14.6.4.1 Detailed Steps/Sequence

5503 The Device shall undertake the validation checks set out in this Section 14.6.4.1 before
 5504 undertaking any other processing of the Command. The validation checks shall be
 5505 undertaken in the sequence laid out. Should a validation check fail, subsequent validation
 5506 checks shall not be undertaken by the Device.

5507 Should any of the checks fail (save for the optional UTRN Check Digit verification), the
 5508 requirements of Section 6.2.4.2 shall apply.

5509 14.6.4.1.1 Verifying the UTRN Check Digit

5510 The Device:

- 5511 • may validate the 20th digit (the UTRN Check Digit) as specified at Section 14.8
 5512 (Calculating and Verifying the UTRN Check Digit); and
- 5513 • shall disregard the 20th decimal digit to determine PPTD prior to undertaking any
 5514 subsequent checks.

5515 14.6.4.1.2 Using the PPTD to calculate the PTUT

5516 PTUT shall take the value of PPTD minus 7,394,156,990,786,306,048.

5517 The Device shall interpret the resulting unsigned integer according to Table 14.6.4.1.2.

PTUT component	Bits
PTUT Sub Class	60-57
PTUT Value Class	56-55
PTUT Value	54-42
PTUT Truncated Originator Counter	41-32
PTUT Supplier MAC	31-0

Table 14.6.4.1.2: Interpretation of the PTUT

5518

5519 *14.6.4.1.3 Verifying PTUT subclass category*

5520 The Device shall carry out the checks specified in Section 14.3.9.

5521 *14.6.4.1.4 Verifying against the maximum credit values*

5522 The Device shall carry out the checks specified in Section 14.3.8.1 and Section 14.3.8.2.

5523 *14.6.4.1.5 Deriving the Originator Counter³⁶*

5524 The Originator Counter shall be derived by:

- 5525 1. creating four 32 bit signed integer variables p, q, r and s;
- 5526 2. setting p = the numeric value of the 10 least significant bits of Highest UTRN Counter;
- 5527 3. setting q = (the numeric value Highest UTRN Counter) – p;
- 5528 4. setting r = the numeric value of PTUT Truncated Originator Counter;
- 5529 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
- 5530 setting $s = r$;
- 5531 6. setting Originator Counter equal to $((q + s) * 2^{32})$

5532 *14.6.4.1.6 Verifying the Originator Counter*5533 The Device shall verify the Originator Counter against the UTRN Counter Cache according
5534 to Section 14.3.6.5535 *14.6.4.1.7 Deriving the Message Identifier*

5536 The Device shall derive the Message Identifier by:

- 5537 • setting the Business Originator ID to the Entity Identifier in the Key Agreement
- 5538 Security Credentials it holds for the Trust Anchor Cell with Remote Party Role as
- 5539 `supplier` and cell usage of `prePaymentTopUp`;
- 5540 • setting the Business Target ID to its own Entity Identifier; and
- 5541 • setting Message Identifier to the concatenation Business Originator ID || Business
- 5542 Target ID || 0x02 || Originator Counter.

5543 *14.6.4.1.8 Validating the PTUT Supplier MAC*

5544 The Device shall validate the PTUT Supplier MAC according to Section 14.3.5.

5545 **14.6.5 Response Construction**

5546 The Device shall first update the UTRN Counter Cache according to Section 14.3.7.

5547 Where the Device is an ESME, the Device shall construct, and send via its HAN interface, a
5548 Response message complying with the requirements of Use Case CS01a, using a Message

³⁶ 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

5549 Identifier as specified in Section 14.6.4.1.7, and where the Originator Counter is as derived
5550 by the calculations in Section 14.6.4.1.5.

5551 Where the Device is a GSME, the Device shall construct, and send via its HAN interface, a
5552 Response message complying with the requirements of Use Case CS01b, using Message
5553 Identifier as specified in Section 14.6.4.1.7, and where the Originator Counter is as derived
5554 by the calculations in Section 14.6.4.1.5.

5555 **14.7 Applying a Prepayment Top Up to an ESME or GSME** 5556 **with consumer entry of a numeric code on a PPMID**

5557 **14.7.1 Description**

5558 This section covers the application of a Prepayment Top Up that has been bought for a
5559 specific GSME or ESME to that GSME or ESME in the case where the consumer enters the
5560 corresponding UTRN on a PPMID on the same SMHAN.

5561 The Use Case covering the Command is referenced in Section 14.7.4.1.2, The Use Case
5562 covering the Response is referenced in Section 14.7.4.1.4.

5563 **14.7.2 Use Case Cross References**

Cross Reference	Value
Grouping	Remote Party Message
Message Type	Command and Responses
Message Type Category	The Command and Response requirements are specifically as detailed in this Section 14.7
Capable of future dated invocation?	No
Protection Against Replay Required?	See Table 14.4.2
SEC User Interface Services Schedule (Service Request) Reference	N/A
Valid Target Device(s)	ESME or GSME
Valid Business Originator role(s) for Command invocation (and so, for DLMS COSEM Commands, which Application Association is to be used for delivery of the Command to the Device) [Remote Party Messages Only]	Supplier
Valid Response Recipient role(s) (only for Messages Authorised by the Access Control Broker on behalf of parties not known to the Device) [Remote Party Messages Only]	N/A
Valid initiating Device type(s) [SMHAN Only Messages]	N/A
Protocol	See this Section 14.7

5564 Table 14.7.2: Use Case Cross References for Prepayment Top Up through PPMID entry

5565 **14.7.3 Pre-conditions**

5566 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 19 decimal digit representation of the PTUT by a process equivalent to the following (Verhoeff's) Algorithm³⁷:

- setting an interim digit (referred to as *IntDig*) to have a value of zero;
- setting an index (referred to as *K*) to have a value of four;
- repeating the following steps with another index (referred to as *J*) taking the nineteen values of the integers from 1 to 19 in succession;
 - setting *CurDig* to the value of the *J*th digit of the 19 decimal digits of the PTUT, where the first digit is the most significant (leftmost as written) and the nineteenth digit the least significant;
 - setting a third index (referred to as *L*) to the value in Table 14.8a using *K* as the Row Index and *CurDig* as the Column Index;
 - if the value of *K* is less than 7, setting *K* to the value of *K*+1, otherwise setting *K* to zero;
- d) setting *IntDig* to the value in Table 14.8b using *IntDig* as the Row Index and *L* as the Column Index;
- setting *IntDig* to the value in row 1 of Table 14.8c using the value of *IntDig* as the Column Index; and
- setting the UTRN Check Digit to the value of *IntDig*.

The UTRN Check Digit may be verified by undertaking exactly the same calculation on the 19 most significant digits of the UTRN, and comparing the result (the final value of *IntDig*, which would be used to set the UTRN Check Digit) to the 20th decimal digit which is the UTRN Check Digit.

		Column Index									
		0	1	2	3	4	5	6	7	8	9
Row Index	0	0	1	2	3	4	5	6	7	8	9
	1	1	5	7	6	2	8	3	0	9	4
	2	5	8	0	3	7	9	6	1	4	2
	3	8	9	1	6	0	4	3	5	2	7
	4	9	4	5	3	1	2	6	8	7	0
	5	4	2	8	6	5	7	3	9	0	1
	6	2	7	9	3	8	0	6	4	1	5
	7	7	0	4	6	9	1	3	2	5	8

Table 14.8a: Setting a third index

		Column Index									
		0	1	2	3	4	5	6	7	8	9
Row	0	0	1	2	3	4	5	6	7	8	9

³⁷ 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

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

5636

15 Message Codes

5637

Message Codes shall be 2 octets in length and shall take the values specified in the 'Use

5638

Case reference' tab in the Mapping Table.

5639

For Messages specified by this GBCS, the most significant bit of the Message Code shall be

5640

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. SMETS and CHTS 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 SMETS and CHTS. It also indicates whether, as per SMETS and CHTS, there is a corresponding Alert issued over the Device's network interface (containing the relevant Event Code). It is important to note that not all Event Codes have a corresponding Alert. Where Alert Code is used elsewhere in this document, it equates to Event Code in Table 16.2.

Alerts sent over the SMHAN are not subject to the same message categorisation as those sent over the WAN. An Alert sent over the SMHAN is a native ZSE message.

16.1.1 Types of Alert

There are two Alert types. All have the same Grouping Header but different payloads as set out below:

- Alert type 1 - Payload comprises Alert Code and Timestamp only (two sub-types: DLMS and ZigBee). These are labelled 'Y(1)' in the 'Alert WAN (Alert type)' column in Table 16.2; and
- Alert type 2 - Payload comprises Alert Code, Timestamp and Use Case specific data as defined in Table 16.2 or main body of document (three sub-types: ASN.1, DLMS and ZigBee). These are labelled 'Y(2)' in the 'Alert WAN (Alert type)' column in Table 16.2.

Table 16.2 sets out the Alert type for each Alert Code. Examples of Use Case specific data include Billing Data Logs and content relating to future dated Commands (e.g. Message ID).

Table 16.2 sets out whether Alerts are mandated, mandatory conditional or non-mandated:

- Mandated - Alerts that Devices must support;
- Mandated conditional – Devices must support at least one from the specified group (e.g. there are seven Alerts in 'mandated – conditional group 1', Devices must support at least one of these seven); and
- Non-mandatory – no requirement for Devices to support, but where implemented Alert Codes shall have the meaning shown in Table 16.2. Further definition of these events may be found in the SSWG specifications³⁸.

16.1.2 Alert Construction

Alert construction is described in the GBCS in a number of places, including:

- Section 7.2.3 details common Message construction for all Alert types;
- Section 7.2.9 details Message construction for Alerts with DLMS COSEM Payloads. Table 7.2.9c details the required components of the Alert;

³⁸ Available from <http://www.triple-3.co.uk/sswg/>.

- 5680 • Section 7.2.10 details Message construction for Alerts with ZSE Payloads. Table
5681 7.2.10c details the required components of the Alert;
- 5682 • Sections 11.2 and 13.3 detail the Message construction for the Alerts with ASN.1
5683 Payload; and
- 5684 • Section 9.2.2 details the Message construction for future dated Alerts.

5685 16.1.3 Event Behaviour

5686 Detail on Event behaviour can be found in SMETS and CHTS using the relevant SMETS
5687 and CHTS reference in Table 16.2.

5688 16.2 Event and Alert Codes

5689 Table 16.2 lists the valid Event and Alert Codes, and sets out their requirements.



GBCS v1.0 Event
and Alert Codes.xlsx

5690
5691 Table 16.2: Event and Alert Codes

5692 16.3 Event Logs

5693 Only GSME, ESME, CHF and GPF have Event Logs. The requirement set out in Table 16.2
5694 to log entries into Event Logs only applies to GSME, ESME, CHF and GPF as follows:

- 5695 • Event Log (GSME, ESME, CHF and GPF);
- 5696 • Security Event Log (GSME, ESME, CHF and GPF);
- 5697 • Power Event Log (ESME); and
- 5698 • ALCS Event Log (ESME).

5699 Use Cases to read logs (all) and clear logs (event logs only) are detailed in the Mapping
5700 Table.

5701 16.4 Requirements

5702 Event / Alert codes shall be 2 octets in length and shall take the values specified in Table
5703 16.2. As per the Device Specifications, all Alerts, Event Log entries, Security Log entries,
5704 Power Event Log entries and ALCS Event Log entries shall contain a UTC date time stamp,
5705 in addition to the Event / Alert code. Non-Critical Alerts can be configured to be sent / not to
5706 be sent using the relevant Commands and Responses defined in Use Cases ECS25a,
5707 ECS25b and GCS20 (all configurable Alerts can be configured in a single Message).

5708 GSME shall reject any ZSE *SetEventConfiguration* command containing an *Event ID* in the
5709 *Event Configuration Payload* with 0x8F in the most significant octet, to ensure Critical Alerts
5710 are always configured on. For clarity, the ESME Alert Configuration Use Cases do not allow
5711 for Alert Codes starting 0x8F.

5712 As specified in Table 16.2 by way of 'x' in a cell, *deviceType* (and for ESME, variant of
5713 ESME) shall determine which Alerts a device shall issue and which Event Log and Security
5714 Log entries it shall record. Where *deviceType* = 0x04 (HCALCS) or 0x05 (PPMID), this
5715 Section 16 only requires the sending of Alerts, since neither Device type is required to have
5716 either an Event Log or a Security Log.

5717 Where an Alert and a Log entry have the same trigger in a Device, the Device shall record
5718 the same UTC date time stamp and the same Event / Alert code in both.

5719 The Remote Party to which an Alert containing a specific Event Code is addressed shall be
 5720 determined by the Remote Party Role as specified in Table 16.2. Where the Remote Party
 5721 Role is stated as Supplier or WAN Provider, the Alert shall be addressed:

- 5722 • to the WAN Provider if `deviceType` = 0x02 (CHF); and
- 5723 • to the Supplier for all other `deviceType` values.

5724 Where a Use Case is specified in Table 16.2 the corresponding Alert shall be constructed
 5725 according to the specified Use Case. Where no Use Case is specified the Alert shall be
 5726 constructed according to Section 7.

5727 Where an Alert has two recipient roles identified, the Device shall place the Entity ID of the
 5728 Supplier in the Business Target ID field and the Entity ID of the other recipient in the
 5729 Supplementary Remote Party ID field.

5730 For any Event Log entries relating to Event Codes 0x8161 and 0x8162, the Device shall
 5731 record the commands input on the User Interface by including the User Interface Command
 5732 Code in the Event Log entry as defined in Table 16.4.

User Interface Command Code	User Interface Command (from SMETS)	GSME	ESME	ESME with ALCS	ESME with Boost Function
0x0001	Activate Boost Period				x
0x0002	Activate Emergency Credit [PIN]	x	x		
0x0005	Add Credit	x	x		
0x0008	Allow Access to User Interface	x	x		
0x000A	Cancel Boost Period				x
0x000B	Check for HAN Interface Commands	x			
0x000C	Disable Privacy PIN Protection [PIN]	x	x		
0x000E	Enable Supply [PIN]	x	x		
0x000F	Extend Boost Period				x
0x0012	Set Privacy PIN [PIN]	x	x		
0x0013	Test Auxiliary Load Control Switch 1			x	
0x0014	Test Auxiliary Load Control Switch 2			x	
0x0015	Test Auxiliary Load Control Switch 3			x	
0x0016	Test Auxiliary Load Control Switch 4			x	
0x0017	Test Auxiliary Load Control Switch 5			x	
0x0018	Test Valve	x			
0x0019	Reset Remaining Battery Capacity	x			
0x001A	Find and Join SMHAN	x	x	x	x

5733 Table 16.4: User Interface Command Codes by Device

5734 For any Event Log entries relating to Event Codes 0x8154 and 0x8155, the Device shall
5735 record the Commands received on the Network Interface by including the Message Code in
5736 the Event Log.

5737 Where a log entry is required to have data additional to the Alert Code and date-time stamp,
5738 that additional data shall be recorded in the 'otherInformation' field of that log entry.

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). In that table, 'A' shall mean that the Remote Party Role shall have access to the Use Case. There shall be no other access allowed by the Device. Remote Party roles align to the Trust Anchor Cells in Section 4.3.2.5. The Access Control Broker controls access for Unknown Remote Parties.

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

18.1.1.1 ZSE Load Control Event command

The ZSE Load Control Event command shall be sent by an ESME, on:

- successful authentication of a Command with Message Code 0x0055;
- to control a HCALCS according to the Auxiliary Load Control Switch 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 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:
 - the duration specified in the Command with Message Code 0x0055;
 - the duration of the command defined in the Auxiliary Load Control Switch Calendar; or
 - the remaining duration calculated as per SMETS.
- the 'Duty Cycle' field set to 0x00, where the Command specifies that the switch is to be turned off; and
- the 'Duty Cycle' field set to 0x64, where the Command specifies that the switch is to be turned on.

The recipient HCALCS shall interpret the value in Duty Cycle accordingly.

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.

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
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

5794 Table 18.1.1.1: ZSE Load Control Event command

5795 **18.1.1.2 Intentionally blank**

5796 **18.1.1.3 ZSE Get Scheduled Events command**

5797 When sending a ZSE Get Scheduled Event command pursuant to section 8.5.2.1 of SMETS,
5798 an HCALCS shall populate that ZCL Command according to Table 18.1.1.3.

5799 On authenticated receipt of ZSE Get Scheduled Event command, the ESME shall send a
5800 ZSE Load Control Event command instructing the HCALCS whether it is to be open or
5801 closed, and for how long it is to be in that state.

5802 On authenticated receipt of ZSE Get Scheduled Event command, the ESME shall send a
 5803 ZSE Load Control Event command instructing the HCALCS whether it is to be open or
 5804 closed, and for how long it is to be in that state, or send a ZSE Default Response command
 5805 with Status = NOT_FOUND if there are no settings for this HCALCS in the ESME's Auxiliary
 5806 Load Control Switch Calendar.

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

5807 Table 18.1.1.3: ZSE Get Scheduled Event command

5808 **18.1.1.4 ZSE Report Event Status command**

Element	Meaning	Value	Octets
ZCL header			
Frame control	Cluster-specific; not manufacturer specific; client-server; allow default response;	0b00000001	1
Transaction sequence number		0x00	1
Command identifier	Report Event Status	0x00	1
ZCL payload			
Issuer Event ID (UINT32)	Set to the event ID from the corresponding ZSE command received from the ESME	See 'Meaning' column	4
Event Status (UINT8)	Refer to ZigBee standard	As per the requirements of this Section 18.1.1	1
Event Status Time (UTCtime)	An HCALCS is not required to have a clock and therefore the HCALC is not required to know UTC time	0x00000001	4
Criticality Level Applied (UINT8)	0x01 = Voluntary	0x01	1
Cooling Temperature Set Point Applied (UINT16)	Not used	0x8000	2

Heating Temperature Set Point Applied (UINT16)	Not used	0x8000	2
Average Load Adjustment Percentage Applied (INT8)	Not used	0x80	1
Duty Cycle Applied (UINT8)	0x00 (0) = switched OFF; 0x64 (100) = switched ON	See 'Meaning' column	1
Event Control (BITMAP8)	Do not randomise	0x00	1
Signature Type (UINT8)	No signature	0x00	1

5809 Table 18.1.1.4: ZSE Report Event Status command

5810 18.2 DLMS COSEM Message Templates

5811 Table 18.2 contains Message Templates for all Use Case with DLMS COSEM payloads.
 5812 These Message Templates are derived from the Mapping Table, and shall be complied with
 5813 in the construction and population of all such Messages.



GBCS v1.0 DLMS
COSEM Message Terr

5814

5815 Table 18.2: DLMS COSEM Message Templates

5816 18.2.1 Encoding

5817 Italicised terms in this Section 18.2.1 shall have their DLMS COSEM meaning.

5818 18.2.1.1 Compact array encoding

5819 The Blue Book definition of *attribute 2* of *Profile Generic* objects may be interpreted as
 5820 requiring '*entry*' to be a *structure* containing a single choice from the DLMS data types. The
 5821 GBCS interprets it as meaning that '*entry*' is a *structure* that can contain multiple choices of
 5822 DLMS data types. These choices vary between instances of Profile Generic object. To
 5823 identify these different structures, the naming convention '*entry_nameOfStructure*' is used.

5824 The GBCS uses the *compact-array* data type in attribute 2 of *Profile Generic* objects. Table
 5825 18.2.1.1 details the derivation of the *contents-description* element within the *compact-array*
 5826 *structure* for the structures used in the *Profile Generic* objects required by this GBCS. These
 5827 encodings are reflected in the DLMS COSEM Message Templates.

<i>Structure definition</i>	<i>Tag</i>	<i>Number of entries (structures and arrays only)</i>	<i>Tag of entries in array</i>	<i>contents-description for compact-array</i>
entry_dIValueLogEntry ::= structure {	0x02	0x02		0x1302020606
timestamp: double-long-unsigned,	0x06			
dIValue: 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			

<i>Structure definition</i>	<i>Tag</i>	<i>Number of entries (structures and arrays only)</i>	<i>Tag of entries in array</i>	<i>contents-description for compact-array</i>
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		0x13020706060100300601000806010008060100080601000806 (single element) or 0x130209060606010030060100040601000806010008060100080601000806 (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		0x13020D060601003006010008060100080601000806010008060505050505 (single element) or 0x13020F0606060100300601000406010008060100080601000806010008060505050505 (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			

<i>Structure definition</i>	<i>Tag</i>	<i>Number of entries (structures and arrays only)</i>	<i>Tag of entries in array</i>	<i>contents-description for compact-array</i>
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		0x130207060505050505
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		0x13020706060100300601000806 010008060100080601000806 (single element) or 0x13020906060601003006010004

<i>Structure definition</i>	<i>Tag</i>	<i>Number of entries (structures and arrays only)</i>	<i>Tag of entries in array</i>	<i>contents-description for compact-array</i>
				0601000806010008060100080601000806 (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_twoDValueLogEntry ::= 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
dIValue: double-long-unsigned,	0x06			
dIValue2: double-long-unsigned	0x06			
}				
entry_alcsLogEntry::= structure {	0x02	0x04		0x13020406121606
timestamp: double-long-unsigned,	0x06			
switchNumberAndAction: long-unsigned,	0x12			
outcome: enum,	0x16			
hANCommandID: double-long-unsigned	0x06			
}				

5828 Table 18.2.1.1: derivation of the *contents-description* element within the *compact-array* structure

5829 **18.2.1.2 Values of the *credit_charge_configuration* attribute of Account (Class ID 111) objects**

5830 There are three SMETS parameters required for all Set Payment Mode Use Cases:

- 5831 • Payment Mode, being Credit or Prepayment;
- 5832 • Suspend Debt Emergency, being True or False and only being relevant when Payment Mode = Prepayment; and
- 5833 • Suspend Debt Disabled, being True or False and only being relevant when Payment Mode = Prepayment.

5834 Note that Disablement Threshold can also be set through the ‘Set Payment Mode to Prepayment’ Use Case.

5835 The combination of these values determines, and is reflected in, the five possible values in the *credit_charge_configuration* attribute of the

5836 Account objects.

5837 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

5838 in Table 18.2.1.2a.

<i>Payment Mode</i>	<i>Suspend Debt Emergency</i>	<i>Suspend Debt Disabled</i>	<i>Value of credit_charge_configuration attribute</i>	<i>Length in octets</i>
Credit	Not relevant	Not relevant	0x0102 020309060000130A00FF09060000130200131400FF0403E0 020309060000130A00FF09060000130200131404FF0403E0	44
Prepayment	True	True	0x010D 020309060000130A00FF09060000130200131400FF0403E0 020309060000130A00FF09060000130200131404FF0403E0 020309060000130A00FF09060000130200131401FF0403C0 020309060000130A00FF09060000130200131402FF0403C0 020309060000130A00FF09060000130200131403FF0403E0 020309060000130A01FF09060000130200131403FF0403E0 020309060000130A01FF09060000130200131400FF0403E0 020309060000130A01FF09060000130200131404FF0403E0 020309060000130A01FF09060000130200131401FF0403C0 020309060000130A01FF09060000130200131402FF0403C0 020309060000130A02FF09060000130200131404FF0403E0 020309060000130A02FF09060000130200131401FF0403E0	275

<i>Payment Mode</i>	<i>Suspend Debt Emergency</i>	<i>Suspend Debt Disabled</i>	<i>Value of credit_charge_configuration attribute</i>	<i>Length in octets</i>
			020309060000130A02FF09060000130200131402FF0403E0	
Prepayment	True	False	0x010D 020309060000130A00FF09060000130200131400FF0403E0 020309060000130A00FF09060000130200131404FF0403E0 020309060000130A00FF09060000130200131401FF0403E0 020309060000130A00FF09060000130200131402FF0403E0 020309060000130A00FF09060000130200131403FF0403E0 020309060000130A01FF09060000130200131403FF0403E0 020309060000130A01FF09060000130200131400FF0403E0 020309060000130A01FF09060000130200131404FF0403E0 020309060000130A01FF09060000130200131401FF0403E0 020309060000130A01FF09060000130200131402FF0403E0 020309060000130A02FF09060000130200131404FF0403E0 020309060000130A02FF09060000130200131401FF0403E0 020309060000130A02FF09060000130200131402FF0403E0	275
Prepayment	False	True	0x010A	212

Payment Mode	Suspend Debt Emergency	Suspend Debt Disabled	Value of credit_charge_configuration attribute	Length in octets
			020309060000130A00FF09060000130200131400FF0403E0 020309060000130A00FF09060000130200131404FF0403E0 020309060000130A00FF09060000130200131401FF0403C0 020309060000130A00FF09060000130200131402FF0403C0 020309060000130A00FF09060000130200131403FF0403E0 020309060000130A01FF09060000130200131403FF0403E0 020309060000130A01FF09060000130200131400FF0403E0 020309060000130A01FF09060000130200131404FF0403E0 020309060000130A01FF09060000130200131401FF0403C0 020309060000130A01FF09060000130200131402FF0403C0	
Prepayment	False	False	0x010A 020309060000130A00FF09060000130200131400FF0403E0 020309060000130A00FF09060000130200131404FF0403E0 020309060000130A00FF09060000130200131401FF0403E0 020309060000130A00FF09060000130200131402FF0403E0 020309060000130A00FF09060000130200131403FF0403E0	212

<i>Payment Mode</i>	<i>Suspend Debt Emergency</i>	<i>Suspend Debt Disabled</i>	<i>Value of credit_charge_configuration attribute</i>	<i>Length in octets</i>
			020309060000130A01FF09060000130200131403FF0403E0 020309060000130A01FF09060000130200131400FF0403E0 020309060000130A01FF09060000130200131404FF0403E0 020309060000130A01FF09060000130200131401FF0403E0 020309060000130A01FF09060000130200131402FF0403E0	

5839 Table 18.2.1.2a: allowable values for the *credit_charge_configuration* attribute for all ESME (except Twin Element variant)

5840 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
5841 18.2.1.2b.

<i>Payment Mode</i>	<i>Suspend Debt Emergency</i>	<i>Suspend Debt Disabled</i>	<i>Value of credit_charge_configuration attribute</i>	<i>Length in octets</i>
Credit	Not relevant	Not relevant	0x0103 020309060000130A00FF09060000130200131400FF0403E0 020309060000130A00FF09060000130200131404FF0403E0 020309060000130A00FF09060000130200131405FF0403E0	65
Prepayment	True	True	0x010F 020309060000130A00FF09060000130200131400FF0403E0 020309060000130A00FF09060000130200131404FF0403E0 020309060000130A00FF09060000130200131401FF0403C0	317

<i>Payment Mode</i>	<i>Suspend Debt Emergency</i>	<i>Suspend Debt Disabled</i>	<i>Value of credit_charge_configuration attribute</i>	<i>Length in octets</i>
			020309060000130A00FF090600000130200131402FF0403 C0 020309060000130A00FF090600000130200131403FF0403 E0 020309060000130A01FF090600000130200131403FF0403 E0 020309060000130A01FF090600000130200131400FF0403 E0 020309060000130A01FF090600000130200131404FF0403 E0 020309060000130A01FF090600000130200131401FF0403 C0 020309060000130A01FF090600000130200131402FF0403 C0 020309060000130A02FF090600000130200131404FF0403 E0 020309060000130A02FF090600000130200131401FF0403 E0 020309060000130A02FF090600000130200131402FF0403 E0 020309060000130A00FF090600000130200131405FF0403 E0 020309060000130A01FF090600000130200131405FF0403 E0	
Prepayment	True	False	0x010F 020309060000130A00FF090600000130200131400FF0403 E0 020309060000130A00FF090600000130200131404FF0403 E0 020309060000130A00FF090600000130200131401FF0403 E0	317

<i>Payment Mode</i>	<i>Suspend Debt Emergency</i>	<i>Suspend Debt Disabled</i>	<i>Value of credit_charge_configuration attribute</i>	<i>Length in octets</i>
			020309060000130A00FF090600000130200131402FF0403E0 020309060000130A00FF090600000130200131403FF0403E0 020309060000130A01FF090600000130200131403FF0403E0 020309060000130A01FF090600000130200131400FF0403E0 020309060000130A01FF090600000130200131404FF0403E0 020309060000130A01FF090600000130200131401FF0403E0 020309060000130A01FF090600000130200131402FF0403E0 020309060000130A02FF090600000130200131404FF0403E0 020309060000130A02FF090600000130200131401FF0403E0 020309060000130A02FF090600000130200131402FF0403E0 020309060000130A00FF090600000130200131405FF0403E0 020309060000130A01FF090600000130200131405FF0403E0	
Prepayment	False	True	0x010C 020309060000130A00FF090600000130200131400FF0403E0 020309060000130A00FF090600000130200131404FF0403E0 020309060000130A00FF090600000130200131401FF0403C0	254

<i>Payment Mode</i>	<i>Suspend Debt Emergency</i>	<i>Suspend Debt Disabled</i>	<i>Value of credit_charge_configuration attribute</i>	<i>Length in octets</i>
			020309060000130A00FF090600000130200131402FF0403 C0 020309060000130A00FF090600000130200131403FF0403 E0 020309060000130A01FF090600000130200131403FF0403 E0 020309060000130A01FF090600000130200131400FF0403 E0 020309060000130A01FF090600000130200131404FF0403 E0 020309060000130A01FF090600000130200131401FF0403 C0 020309060000130A01FF090600000130200131402FF0403 C0 020309060000130A00FF090600000130200131405FF0403 E0 020309060000130A01FF090600000130200131405FF0403 E0	
Prepayment	False	False	0x010C 020309060000130A00FF090600000130200131400FF0403 E0 020309060000130A00FF090600000130200131404FF0403 E0 020309060000130A00FF090600000130200131401FF0403 E0 020309060000130A00FF090600000130200131402FF0403 E0 020309060000130A00FF090600000130200131403FF0403 E0 020309060000130A01FF090600000130200131403FF0403 E0	254

Payment Mode	Suspend Debt Emergency	Suspend Debt Disabled	Value of credit_charge_configuration attribute	Length in octets
			020309060000130A01FF09060000130200131400FF0403E0	
			020309060000130A01FF09060000130200131404FF0403E0	
			020309060000130A01FF09060000130200131401FF0403E0	
			020309060000130A01FF09060000130200131402FF0403E0	
			020309060000130A00FF09060000130200131405FF0403E0	
			020309060000130A01FF09060000130200131405FF0403E0	

5842 Table 18.2.1.2b: allowable values for the *credit_charge_configuration* attribute for all ESME (Twin Element variant)

5843 **18.2.1.3 Deriving the values of the *credit_charge_configuration* attribute of Account (Class ID 111) objects – informative**

5844 This section explains the derivation of the five values of this attribute that an ESME can accept.

5845 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

Component	Hex value	Length in octets	Notes
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	

5846 Table 18.2.1.3a: *credit_charge_configuration* attribute encoding

5847 So the value of the *credit_charge_configuration_element* attribute is a 21 octet long concatenation:

5848 0x02030906 || credit object OBIS code || 0x0906 || charge object OBIS code || 0x0403 || collection bit string || 0b00000

5849 The meaning of each *credit_charge_configuration_element* is that this charge can be collected from this credit object, except in possible meter
5850 states specified by the *collection_configuration* bit string.

5851 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

5852 Table 18.2.1.3b: Class 112 Credit objects

5853 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
5854 relevant in Credit Mode.

SMET Reference Component	OBIS Code (decimal)	OBIS Code (hexadecimal)	Payment Mode
DebtRecoveryRates[1]	0-0:19.20.1.255	0x000013020013 1401FF	Prepayment
DebtRecoveryRates[2]	0-0:19.20.2.255	0x000013020013 1402FF	Prepayment
DebtRecoveryPerPayment	0-0:19.20.3.255	0x000013020013 1403FF	Prepayment
SecondaryTariffTOUPriceMatrix (Twin element ESME only)	0-0:19.20.5.255	0x000013020013 1405FF	Prepayment and Credit
StandingCharge	0-0:19.20.4.255	0x000013020013 1404FF	Prepayment and Credit
TariffBlockPriceMatrixTOU	0-0:19.20.0.255	0x000013020013 1400FF	Prepayment and Credit

5855 Table 18.2.1.3c: Class 113 Charge objects

5856 As defined in the Blue Book, the *collection_configuration* bit string determines whether a charge is collected from a credit dependent on ESME
5857 state.

5858 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
5859 limiting periods).

5860 In Credit Mode, collection continues in all states, so the value of all three bits is always 0b1.

5861 In Prepayment Mode, *collection_configuration* is set according to Suspend Debt Disabled (affects Bit 0) values, and the pairing of charge and
5862 credit object.

5863 Suspend Debt Emergency being True means that DebtRecoveryRates[1..2] and StandingCharge are collected from AccumulatedDebt rather
5864 than EmergencyCreditBalance, when Emergency Credit is in use, so Suspend Debt Emergency is specified by way of pairing charge and credit
5865 objects accordingly. Note that Bit 2 of *collection_configuration* shall always be fixed at 0b1.

5866 Suspend Debt Disabled being True means that DebtRecoveryRates[1..2] are no longer collected when the supply is disabled due to lack of
5867 credit.

5868 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	0x0000130200131400FF (TariffBlockPriceMatrixTOU)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130200131400FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130200131404FF (StandingCharge)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130200131404FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130200131405FF (SecondaryTariffTOUPriceMatrix (Twin element ESME only))	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130200131405FF0403E0

5869 Table 18.2.1.3d: Class 113 Charge objects

5870 When the mode is set as in Table 18.2.1.3e:

Payment Mode	Suspend Debt Emergency	Suspend Debt Disabled
Prepayment	False	False

5871 Table 18.2.1.3e: Prepayment states

5872 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	0x0000130200131400FF (TariffBlockPriceMatrixTOU)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130200131400FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130200131404FF (StandingCharge)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130200131404FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130200131401FF (DebtRecoveryRates[1])	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130200131401FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130200131402FF (DebtRecoveryRates[2])	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130200131402FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130200131403FF (DebtRecoveryPerPayment)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130200131403FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130200131403FF (DebtRecoveryPerPayment)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130200131403FF0403E0

Tag & Length	Credit object	Tag & Length	Charge object	Tag & Length	Collection bit string	trailing bits	Array entry
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130200131400FF (TariffBlockPriceMatrixTOU)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130200131400FF 0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130200131404FF (StandingCharge)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130200131404FF 0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130200131401FF (DebtRecoveryRates[1])	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130200131401FF 0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130200131402FF (DebtRecoveryRates[2])	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130200131402FF 0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130200131405FF (SecondaryTariffTOUPriceMatrix (Twin element ESME only))	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130200131405FF 0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130200131405FF (SecondaryTariffTOUPriceMatrix (Twin element ESME only))	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130200131405FF 0403E0

5873 Table 18.2.1.3f: *credit_charge_configuration_element* array entries

5874 Note that, as per SMETS, the value of MeterBalance determines whether charges are collected from EmergencyCreditBalance or
5875 MeterBalance.

5876 When the mode is set as in Table 18.2.1.3g:

Payment Mode	Suspend Debt Emergency	Suspend Debt Disabled
Prepayment	False	True

5877 Table 18.2.1.3g: Prepayment states

5878 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	0x0000130200131400FF (TariffBlockPriceMatrixTOU)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130200131400FF 0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130200131404FF (StandingCharge)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130200131404FF 0403E0

Tag & Length	Credit object	Tag & Length	Charge object	Tag & Length	Collection bit string	trailing bits	Array entry
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130200131401FF (DebtRecoveryRates[1])	0x0403	0b110 (do not collect when supply is disabled due to no credit)	0b00000	0x020309060000130A00FF09060000130200131401FF0403C0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130200131402FF (DebtRecoveryRates[2])	0x0403	0b110 (do not collect when supply is disabled due to no credit)	0b00000	0x020309060000130A00FF09060000130200131402FF0403C0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130200131403FF (DebtRecoveryPerPayment)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130200131403FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130200131403FF (DebtRecoveryPerPayment)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130200131403FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130200131400FF (TariffBlockPriceMatrixTOU)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130200131400FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130200131404FF (StandingCharge)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130200131404FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130200131401FF (DebtRecoveryRates[1])	0x0403	0b110 (do not collect when supply is disabled due to no credit)	0b00000	0x020309060000130A01FF09060000130200131401FF0403C0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130200131402FF (DebtRecoveryRates[2])	0x0403	0b110 (do not collect when supply is disabled due to no credit)	0b00000	0x020309060000130A01FF09060000130200131402FF0403C0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130200131405FF (SecondaryTariffTOUPriceMatrix (Twin element ESME only))	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130200131405FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130200131405FF (SecondaryTariffTOUPriceMatrix (Twin element ESME only))	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130200131405FF0403E0

5879 Table 18.2.1.3h: *credit_charge_configuration_element* array entries

5880 When the mode is set as in Table 18.2.1.3i:

<i>Payment Mode</i>	<i>Suspend Debt Emergency</i>	<i>Suspend Debt Disabled</i>
Prepayment	True	False

5881 Table 18.2.1.3i: Prepayment states

5882 the *credit_charge_configuration_element* array entries are as per Table 18.2.1.3j.

<i>Tag & Length</i>	<i>Credit object</i>	<i>Tag & Length</i>	<i>Charge object</i>	<i>Tag & Length</i>	<i>Collection bit string</i>	<i>trailing bits</i>	<i>Array entry</i>
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130200131400FF (TariffBlockPriceMatrixTOU)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130200131400FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130200131404FF (StandingCharge)	0x0403	0b111 (collectable in all circumstances)	0b0000	0x020309060000130A00FF09060000130200131404FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130200131401FF (DebtRecoveryRates[1])	0x0403	0b111 (collectable in all circumstances)	0b0000	0x020309060000130A00FF09060000130200131401FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130200131402FF (DebtRecoveryRates[2])	0x0403	0b111 (collectable in all circumstances)	0b0000	0x020309060000130A00FF09060000130200131402FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130200131403FF (DebtRecoveryPerPayment)	0x0403	0b1110 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130200131403FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130200131403FF (DebtRecoveryPerPayment)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130200131403FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130200131400FF (TariffBlockPriceMatrixTOU)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130200131400FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130200131404FF (StandingCharge)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130200131404FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130200131401FF (DebtRecoveryRates[1])	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130200131401FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130200131402FF (DebtRecoveryRates[2])	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130200131402FF0403E0
0x02030906	0x0000130A02FF (AccumulatedDebt)	0x0906	0x0000130200131404FF (StandingCharge)	0x0403	0b111 (collect in Emergency Credit period –	0b00000	0x020309060000130A02FF09060000130200131404FF0403E0

Tag & Length	Credit object	Tag & Length	Charge object	Tag & Length	Collection bit string	trailing bits	Array entry
					see note at bottom of table)		
0x02030906	0x0000130A02FF (AccumulatedDebt)	0x0906	0x00000130200131401FF (DebtRecoveryRates[1])	0x0403	0b111 (collect in Emergency Credit period – see note at bottom of table)	0b00000	0x020309060000130A02FF0906000130200131401FF 0403E0
0x02030906	0x0000130A02FF (AccumulatedDebt)	0x0906	0x00000130200131402FF (DebtRecoveryRates[2])	0x0403	0b111 (collect in Emergency Credit period – see note at bottom of table)	0b00000	0x020309060000130A02FF0906000130200131402FF 0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x00000130200131405FF (SecondaryTariffTOUPriceMatrix (Twin element ESME only))	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF0906000130200131405FF 0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x00000130200131405FF (SecondaryTariffTOUPriceMatrix (Twin element ESME only))	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF0906000130200131405FF 0403E0

5883 Table 18.2.1.3j: *credit_charge_configuration_element* array entries

5884 Note that, as per SMETS, charges shall only accrue to AccumulatedDebt in Emergency Credit periods.

5885 When the mode is set as in Table 18.2.1.3k:

Payment Mode	Suspend Debt Emergency	Suspend Debt Disabled
Prepayment	True	True

5886 Table 18.2.1.3k Prepayment states

5887 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	0x00000130200131400FF (TariffBlockPriceMatrixTOU)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF0906000130200131400FF 0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x00000130200131404FF (StandingCharge)	0x0403	0b111	0b00000	0x020309060000130A00FF0906000130200131404FF 0403E0

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	0x0000130200131401FF (DebtRecoveryRates[1])	0x0403	0b110 (do not collect when supply is disabled due to no credit)	0b00000	0x020309060000130A00FF09060000130200131401FF0403C0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130200131402FF (DebtRecoveryRates[2])	0x0403	0b110 (do not collect when supply is disabled due to no credit)	0b00000	0x020309060000130A00FF09060000130200131402FF0403C0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x0000130200131403FF (DebtRecoveryPerPayment)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF09060000130200131403FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130200131403FF (DebtRecoveryPerPayment)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130200131403FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130200131400FF (TariffBlockPriceMatrixTOU)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130200131400FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130200131404FF (StandingCharge)	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF09060000130200131404FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130200131401FF (DebtRecoveryRates[1])	0x0403	0b110 (do not collect when supply is disabled due to no credit)	0b00000	0x020309060000130A01FF09060000130200131401FF0403C0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x0000130200131402FF (DebtRecoveryRates[2])	0x0403	0b110 (do not collect when supply is disabled due to no credit)	0b00000	0x020309060000130A01FF09060000130200131402FF0403C0
0x02030906	0x0000130A02FF (AccumulatedDebt)	0x0906	0x0000130200131404FF (StandingCharge)	0x0403	0b111 (collect in Emergency Credit period – see note at bottom of table)	0b00000	0x020309060000130A02FF09060000130200131404FF0403E0
0x02030906	0x0000130A02FF (AccumulatedDebt)	0x0906	0x0000130200131401FF (DebtRecoveryRates[1])	0x0403	0b111 (collect in Emergency	0b00000	0x020309060000130A02FF09060000130200131401FF0403E0

Tag & Length	Credit object	Tag & Length	Charge object	Tag & Length	Collection bit string	trailing bits	Array entry
					Credit period – see note at bottom of table)		
0x02030906	0x0000130A02FF (AccumulatedDebt)	0x0906	0x00000130200131402FF (DebtRecoveryRates[2])	0x0403	0b111 (collect in Emergency Credit period – see note at bottom of table)	0b00000	0x020309060000130A02FF0906000130200131402FF0403E0
0x02030906	0x0000130A00FF (MeterBalance)	0x0906	0x00000130200131405FF (SecondaryTariffTOUPriceMatrix (Twin element ESME only))	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A00FF0906000130200131405FF0403E0
0x02030906	0x0000130A01FF (EmergencyCreditBalance)	0x0906	0x00000130200131405FF (SecondaryTariffTOUPriceMatrix (Twin element ESME only))	0x0403	0b111 (collectable in all circumstances)	0b00000	0x020309060000130A01FF0906000130200131405FF0403E0

5888 Table 18.2.1.3l: *credit_charge_configuration_element* array entries

5889 Note that, as per SMETS, charges shall only accrue to AccumulatedDebt in Emergency Credit periods.

5890 **18.2.1.4 Encoding of Billing Calendar start date-time and periodicity**

5891 Table 18.2.1.4 sets out how the components of the Billing Calendar start date-time and periodicity should be encoded.

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
Length	0x02	1	2 elements in structure
Time			
Tag	0x09	1	Tag for structure
Length	0x04	1	4 octets in DLMS encoded time
Value	See note	4	Time part of the start date-time, as per section 4.1.6.1 of the Blue Book

Component	Hex value	Length in octets	Notes
Date			
Tag	0x09	1	5 octets in DLMS encoded date
Length	0x05	1	2 elements in structure
Value			
year highbyte,	0xFF	1	0xFF means not specified
year lowbyte,	0xFF	1	0xFF means not specified
month,	0xFF	1	0xFF means not specified
day of month,	0xFF unless periodicity is monthly. If periodicity is monthly, this shall be the day of the month of the start date-time.	1	0xFF means not specified.
day of week	0xFF unless periodicity is weekly If periodicity is weekly, this shall be the day of the week of the start date-time.	1	0xFF means not specified

5892 Table 18.2.1.4: Encoding of Billing Calendar start date-time and periodicity

5893 18.3 Illustrative command and response instantiation and DER encoding

5894 18.3.1 Illustrative @UpdateSecurityCredentials.CommandPayload instantiation and its DER encoding – 5895 informative

5896 supplierUpdatingAllSupplierCertificates in Table 18.3.1a is an ASN.1 structured value assignment. This specific example is
5897 where a Device's Supplier is instructing the Device to replace both the Supplier Digital Signing and Key Agreement credentials on the Device,
5898 and resetting Protection Against Replay counters. In business terms, an example of this would be at Change of Supplier.

5899 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
5900 values.

ASN.1	Notes
supplierUpdatingAllSupplierCertificates CommandPayload ::=	
{authorisingRemotePartyControl	

ASN.1	Notes
<pre> {credentialsReplacementMode <i>supplierBySupplier</i>, authorisingRemotePartyTACellIdentifier {trustAnchorCellRemotePartyRole <i>supplier</i>, trustAnchorCellKeyUsage { <i>digitalSignature</i>}}, authorisingRemotePartySeqNumber <i>123456789</i>, newRemotePartyFloorSeqNumber <i>987654321</i>} replacements {{replacementCertificate <i>'0A7C8E9F123456789ABCDEF01234'H</i>, targetTrustAnchorCell {trustAnchorCellRemotePartyRole <i>supplier</i>, trustAnchorCellKeyUsage { <i>digitalSignature</i>}}} {replacementCertificate <i>'0B34269F123456789ABCDEF01234'H</i>, targetTrustAnchorCell {trustAnchorCellRemotePartyRole <i>supplier</i>, trustAnchorCellKeyUsage {<i>keyAgreement</i>}}}} certificationPathCertificates { <i>'FFAABB9F123456789ABCDEF01234'H</i> }} </pre>	<p>This message is for the supplier replacing supplier credentials</p> <p>The public key to be used to check the signature on this message is the supplier digital signing key currently held by the Device.</p> <p>This is the existing supplier's counter, so greater than any this supplier has used</p> <p>This is the new supplier's counter, which the Device should use if the Command is successful</p> <p>The new supplier's digital signing certificate ...</p> <p>... which is to be placed in the Device's supplier, digital signature Trust Anchor Cell</p> <p>The new supplier's key agreement certificate...</p> <p>which is to be placed in the Device's supplier, key agreement Trust Anchor Cell</p> <p>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.</p>

5901 Table 18.3.1a: Illustrative @UpdateSecurityCredentials.CommandPayload instantiation – ASN.1 structure

5902 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
5903 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 Protection Against Replay 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 Protection Against Replay 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

5904 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
<pre>supplierUpdatingAllSupplierCertificatesResponse ResponsePayload ::= { commandAccepted NULL, executionOutcome {authorisingRemotePartySeqNumber 123456789, credentialsReplacementMode <i>supplierBySupplier</i>, remotePartySeqNumberChanges {{otherRemotePartyRole <i>supplier</i>, otherRemotePartyFloorSeqNumber <i>987654321</i>}} }, replacementOutcomes {{ {affectedTrustAnchorCell { trustAnchorCellRemotePartyRole <i>supplier</i>, trustAnchorCellKeyUsage { <i>digitalSignature</i>}}, statusCode <i>success</i>, existingSubjectUniqueID '123456789ABCDEF0'H, existingSubjectKeyIdentifier '1234567890123456'H, replacingSubjectUniqueID 'FEDCBA9876543210'H,</pre>	<p>The corresponding Command was for the Supplier replacing supplier credentials</p> <p>This is the new supplier's counter, which the Device will now use for Protection Against Replay in relation to the supplier role</p> <p>This outcome is for the supplier digital signing store</p> <p>The old supplier's Entity Identifier</p> <p>The KeyIdentifier for the old supplier's digital signing key</p> <p>The new supplier's Entity Identifier</p> <p>The KeyIdentifier for the old supplier's digital signing key</p> <p>This outcome is for the supplier key agreement store</p>

ASN.1	Notes
<pre> replacingSubjectKeyIdentifier 'ABCDEABCDEABCDEA'H}, {affectedTrustAnchorCell {trustAnchorCellRemotePartyRole supplier, trustAnchorCellKeyUsage { keyAgreement}}, statusCode success, existingSubjectUniqueID '123456789ABCDEF0'H, existingSubjectKeyIdentifier '0987654321098765'H, replacingSubjectUniqueID 'FEDCBA9876543210'H, replacingSubjectKeyIdentifier 'FEDCBFEDCBFEDCBF'H}}}} </pre>	

5911 Table 18.3.2a: Illustrative @UpdateSecurityCredentials.Response instantiation – ASN.1 structure

5912 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
5913 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 Protection Against Replay 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 Protection Against Replay counter in hexadecimal
replacementOutcomes SEQUENCE OF:		
tag = [UNIVERSAL 16] constructed;	0X30	Tag for SEQUENCE
length =	0X6C	Length of 108
content =		
ReplacementOutcome SEQUENCE:		
tag = [UNIVERSAL 16] constructed;	0X30	Tag for SEQUENCE
length =	0X34	Length of 52
content =		

Component	Value	Notes
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 =	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

Component	Value	Notes
content =	0XABCDEABCDEABCDEA	KeyIdentifier
ReplacementOutcome SEQUENCE:		
tag = [UNIVERSAL 16] constructed;	0X30	Tag for SEQUENCE
length =	0X34	
content =		
affectedTrustAnchorCell TrustAnchorCellIdentifier SEQUENCE:		
tag = [UNIVERSAL 16] constructed;	0X30	Tag for SEQUENCE
length =	0X07	
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

Component	Value	Notes
length =	0X08	8 octet length of Entity Identifier
content =	0XFEDCBA9876543210	
replacingSubjectKeyIdentifier OCTET STRING:		
tag = [UNIVERSAL 4] primitive;	0X04	Tag for OCTET STRING
length =	0X08	length of KeyIdentifier
content =	0XFEDCBFEDCBFEDCBF	KeyIdentifier

5914 Table 18.3.2b: Illustrative @UpdateSecurityCredentials.ResponsePayload instantiation – DER encoding

5915 18.4 Cryptographic Test Vectors

5916 This Section 18.4 provides cryptographic calculations in relation to a number of sample messages. The sample messages' contents align with
5917 the corresponding Message Templates in Section 18.2. To undertake cryptographic calculations, a number of details about the Smart Metering
5918 Entities involved are also required, not least Key Pairs, Entity Identifiers and Originator Counters. This section specifies and uses sample
5919 values of such attributes.

5920 18.4.1 Cryptographic Calculations

5921 Create details for three Smart Metering Entities with associated Keys and shared secrets:

5922

5923 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

5924

5925 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

5926

5927 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

5928

5929 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:	0x09:01:00:00:00:00:00:00:01:08:12:34:56:78:9A: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
--The per message secret number:	60336962327539050191752715802083620799857562569756173778554746475842619617775
--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:09:01:00:00:00:00:00:00:01:08:12:34:56:78:9A:BC:DE:F0:08:FF:FF:FF:FF:FF:FF:FF:FE:00:02:00:B3:35
--All of the Message parts covered by the general-signing structure	0xDF:09:01:00:00:00:00:00:00:01:08:12:34:56:78:9A: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 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:09:01:00:00:00:00:00:00:01:08:12:34:56:78:9A: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 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: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

5930

5931 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

5932

5933 Supplier A has now increased its Originator Counter by 1.

5934

5935 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: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: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

5936

5937 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:09:02:00:00:00:00:00:00:02:08: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:09:02:00:00:00:00:00:00:02:12:34:56:78:9A: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:09:02:00:00:00:00:00:00:02:08:FF:FF:FF:FF:FF:FF:FE:08:12:34:56:78:9A:BC:DE:F0:00:02:00:22:0C:DA:20:00:00:02:00:00:01:00:01:02:00:00
--The 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

5938

5939

18.4.2 Example Messages Produced

5940

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	0x00000000000000	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	0xFFFFFFFFFFFFFFFFFE	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	0x59045AB0F554622B0360340B9D87B5A23ED5723B41DE3 F206E58CDD10F915B9FE2E12E2DA3632478A8DF678E4188 95869AC1E55318CCE04D120D2D6B44DC167B	64
____mac-content	0x5D832D15B57A56D620F198B3	12

5941

5942 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	0xFFFFFFFFFFFFFFFFFE	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	0xA01AB69ED8A6566AB11643C7358260A88A8B60978593E 76A4E93193585D98D9BAD843878F22E794B53F6F280F9F1 C848D9D38FC050CDDD58827563E2B0FA19E2	64

5943

5944 ECS12 Set Change of Tenancy date on ESME

5945

5946 Command Message Structure

Name	Encoded Content	Encoded Length
MAC Header (general-ciphering)		
____tag	0xDD	1
____contents	0x00000000000000	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	0xFFFFFFFFFFFFFFFFFE	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

5947

5948 Response Message Structure

Name	Encoded Content	Encoded Length
MAC Header (general-ciphering)		
____tag	0xDD	1
____contents	0x00000000000000	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

5949

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.

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 Message Code (see Section 15)
Use Case Cross References	See Section 19.1.1
Use Case Access Permissions	A summary of User Roles that can perform the Use Case. See Section 17 for Remote Party Usage Rights and Section 4.3.2.6 for Trust Anchor Cells applicable. Note that Use Cases from Unknown Remote Parties are performed using the Remote Party Role of Access Control Broker
SMETS / CHTS Objects applicable to Use Case	A list of SMETS / CHTS attributes and associated methods that are applicable to the Use Case. This confirms the properties required by SMETS / CHTS for the attribute/method. This also provides information on the User Interface Service Request 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
Requires protection against replay?	Yes / No	Needed to identify which GBCS requirements apply. See Section 4.3.1.5

Cross Reference	Possible Values	Notes
SEC User Interface Services Schedule (Service Request) Reference	[e.g. 6.20 SetDeviceConfiguration(MPxN)]	Traceability to SEC-listed DCC Service Requests
Read Or Update	Read, Update	Identifies whether the purpose of the Use Case is 'Read' or 'Update'
Response Recipient different from Command Sender	Yes or Blank	Identifies where a Response is sent to a different Remote Party than the originator of the associated Service Request
Use Case Access Permissions	Supplier (C) Supplier (NC) Supplier prepay top up Network Operator (C) Network Operator (NC) Access Control Broker (NC) WAN Provider (C) Access Control Broker (C)	Lists which Remote Party Roles may originate the Command within the Use Case. This separates (C) critical and (NC) non critical See Section 17 for more details

5962 Table 19.1.1: Allowable values for SMETS / CHTS Use Case Cross References

5963 19.1.2 Objects Applicable to Use Case Section

5964 This section in Table 19.3 contains a 'SMETS Objects applicable to Use Case' table to
5965 provide traceability between SMETS functions and methods and the Use Case.

5966 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

5967 Table 19.1.2: SMETS objects applicable to Use Case

5968 19.1.3 Pre-conditions

5969 Pre-conditions represent conditions for which Device logic is required to ensure correct
5970 operation of commands contained within a message, on the Device. Exception conditions
5971 (such as failures) that are managed by the Protocol are not captured as Pre-conditions.
5972 Manufacturers of Devices must only enforce Pre-conditions that are stated in the Use Cases.
5973 Note that the use of Pre-conditions is minimised in favour of controls being implemented on
5974 Service User systems.

5975 19.1.4 Actions

5976 Actions stipulate additional Device actions that must be performed together with successful
5977 execution of the Use Case.

5978 19.2 Use Case-specific content

5979 Each Use Case is given a unique reference and a title.

5980 19.2.1 DLMS COSEM specific content

5981 Table 19.2.1 sets out the Use Case specific attributes and methods and the DLMS specific
5982 mapping.

5983 Within any DLMS COSEM Payload, cosem-attribute-descriptors and cosem-method-
5984 descriptors, and associated fields, shall be ordered based on the contents of columns in the
5985 Mapping Table. The sort order, described by columns headings in the Mapping Table, shall
5986 be:

- 5987 • first by DLMS Sequence;
- 5988 • then by DLMS: Class;
- 5989 • then by OBIS Code;
- 5990 • then by Attribute (A) / Method (M); and
- 5991 • then by Attribute / Method Number.

5992 The sort order shall be ascending in all cases.

5993 For structures, the sequence of elements within a structure shall be as defined in the Blue
5994 Book, where it is defined in the Blue Book, or as per the Mapping Table, where it is not
5995 defined in the Blue Book.

5996 For clarity, the SMETS / CHTS table in each Use Case is sorted in this same order.

Row Name	Meaning
Mapping table row #	Identifier of the SMETS / CHTS object's row in the Mapping Table
SMETS / CHTS Ref	The section(s) with SMETS/CHTS that refer to the SMETS / CHTS name
SMETS / CHTS Name	The attribute name as specified in SMETS / CHTS
Class	Denotes the DLMS Interface Class
OBIS Code	defines identification codes for all data in DLMS / COSEM compliant metering equipment
Attribute or Method	Denotes whether the row relates to an (A)tttribute or (M)ethod
Attribute / Method Number	Forms part of the attribute identity.
Attribute / Method Name and Blue Book reference	The name given to the DLMS object
DLMS COSEM Data type	The data type assigned to the DLMS object
Constant Value	Where this field is present, this is a fixed value for the life of the Device
Notes	Additional useful information

5997 Table 19.2.1: DLMS mapping for Use Case specific attributes / methods

5998 19.2.2 ZSE specific content

5999 Table 19.3 provides information on the ZSE commands required successfully to complete
6000 the Use Case. These must be processed in the order listed in Table 19.2.2.

6001 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

6002 Table 19.2.2: ZSE specific content

6003 19.3 Embedded Use Cases

6004 Table 19.3 contains the Use Cases that fulfil the interface requirements to cover Commands
6005 (and their Responses) and Alerts (where applicable). In addition, it includes ZSE Message
6006 Templates.

6007 Note: DLMS COSEM methods that have values which have an impact on the execution of
6008 the method (that is, methods with input values that are not integer(0)), the DLMS part of the
6009 Mapping Table and the Use Case include two or more rows. One row contains the method,
6010 and the subsequent row(s) contain the value(s) to be sent with the method.

6011 A number of Use Cases are also covered in GBCS main body. These are identifiable from
6012 the Table of Contents.



GBCS v1_0 Use
Cases.html

6013

6014 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 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.



GBCS v1.0 Mapping
Table.xlsm

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.

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.

- 6069 **Authenticated Decryption**
- 6070 Has the same meaning as specified in *NIST Special Publication 800-38D*:
 6071 <http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf>
- 6072 **Authenticated Encryption (AE)**
- 6073 Has the same meaning as specified in *NIST Special Publication 800-38D*:
 6074 <http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf>
- 6075 **Authentication**
- 6076 The method used to confirm the identity of entities or Devices wishing to communicate and
 6077 'Authenticated' and 'Authenticity' shall be construed accordingly.
- 6078 **Authentication Key**
- 6079 Shall be as defined in the Green Book.
- 6080 **Authorisation**
- 6081 The process of granting access to a resource and 'Authorised' shall be construed
 6082 accordingly.
- 6083 **Authorised Public Key Infrastructure (APKI)**
- 6084 A key infrastructure that is compliant with the Certificate related requirements of this GBCS.
- 6085 **Auxiliary Load Control Switch**
- 6086 A switch or other means of controlling a load on the Supply.
- 6087 **Blue Book**
- 6088 The DLMS Blue Book Version *DLMS UA 1000-1 Ed. 12.0*. This document can be obtained
 6089 from the DLMS User Association: <http://www.dlms.com>.
- 6090 **Boost Function**
- 6091 ESME functionality described in SMETS.
- 6092 **Break On Error**
- 6093 Shall have the Green Book meaning of Break On Error used in relation to
 6094 'Processing_Option'.
- 6095 **Business Originator**
- 6096 The Smart Metering Entity sending the first Message in a Use Case.
- 6097 **Business Target**
- 6098 The Smart Metering Entity receiving the first Message in a Use Case.
- 6099 **Certificate**
- 6100 An electronic document that binds an identity, and possibly other information, to a Public
 6101 Key.
- 6102 **Certification Request**
- 6103 A message requesting the issue of a Certificate by a Certification Authority.
- 6104 **Certification Authority (CA)**
- 6105 A trusted entity which issues Certificates.
- 6106 **Certification Authority Certificate**
- 6107 A Certificate issued to a Certification Authority that allows Certification Path Validation in
 6108 relation to Remote Party's Certificates.

- 6109 [Certification Path Validation](#)
- 6110 Shall have the meaning defined in Section 4.3.2.8.
- 6111 [Certification Revocation List \(CRL\) Validation](#)
- 6112 Shall have the meaning defined in Section 4.3.2.8.
- 6113 [Ciphred Information](#)
- 6114 Shall have the meaning defined in Section 8.4.
- 6115 [Ciphertext](#)
- 6116 An output of the Authenticated Encryption function and an input of the Authenticated
6117 Decryption function defined in *NIST Special Publication 800-38D*. The unencrypted form of
6118 the Ciphertext is the Plaintext.
- 6119 [Clock](#)
- 6120 A timing mechanism which has a minimum resolution of 1 second.
- 6121 [Command](#)
- 6122 An instruction to perform a function received or sent by any interface.
- 6123 [Command Response Alert \(CRA\) Flag](#)
- 6124 An element within a Message Header that enumerates whether the Message is a Command
6125 or a Response or an Alert.
- 6126 [Commercial Product Assurance Security Characteristic](#)
- 6127 The security characteristics for the relevant Device as indicated in Section 1.0.
- 6128 [Communications Hub](#)
- 6129 A Device complying with the CHTS.
- 6130 [Communications Hub Function \(CHF\)](#)
- 6131 The functionality in the Communications Hub specific to its operation as a bridge between
6132 the WAN Interface and the HAN Interface.
- 6133 [Communications Hub Technical Specifications \(CHTS\)](#)
- 6134 Communications Hub Technical Specifications set out in Schedule 10 of the Smart Energy
6135 Code.
- 6136 [Confidentiality](#)
- 6137 The state of information, in transit or at rest, where there is assurance that it is not
6138 accessible by Unauthorised parties through either unintentional means or otherwise.
- 6139 [Consumer](#)
- 6140 A person who lawfully resides at the Premises that is being Supplied.
- 6141 [Consumer Access Device \(CAD\)](#)
- 6142 A Device which, in terms of this GBCS, supports the same Messages as an IHD.
- 6143 [Consumption](#)
- 6144 In the context of GSME, Gas Consumption or in the context of ESME, Electricity
6145 Consumption information.
- 6146 [Contingency Key](#)
- 6147 A feature of Trust Anchor Management Protocol (RFC 5934), and only ever used in a
6148 recovery scenario when the `root` Certificate (Apex Trust Anchor) needs to be replaced.

6149 **Critical Message**

6150 A Remote Party Message which may relate to supply being affected, financial fraud or the
6151 compromise of Device security. Critical, Critical Commands, Critical Alerts and Critical
6152 Responses shall be construed accordingly.

6153 **Cryptographic Algorithm**

6154 An algorithm for performing one or more cryptographic functions which may include
6155 Encryption; Decryption; digitally signing or Hashing of information, data, or messages; or
6156 exchange of Security Credentials.

6157 **Cryptographic Protection**

6158 A part of a Message constructed to provide assurance to the Message recipient in terms of
6159 one or more of integrity, authenticity, non-repudiation and Confidentiality.

6160 **Currency Units**

6161 The units of monetary value in major and minor units.

6162 **Current Private Key**

6163 A Device Private Key for which the Device has successfully received and processed a
6164 Certificate for the corresponding Public Key as defined in Section 13.5.

6165 **Data and Communications Company (DCC)**

6166 The holder of the licence for the provision of a smart meter communication service granted
6167 pursuant to section 6(1)(f) or 6(1A) of the Electricity Act 1989 or section 7AB of the Gas Act
6168 1986.

6169 **Data Store**

6170 An area of a Device capable of storing information for future retrieval.

6171 **Decryption**

6172 The process of converting Encrypted information by an Authorised party to recover the
6173 original information. Like terms shall be construed accordingly.

6174 **Device**

6175 A Device that is one of ESME, GSME, Gas Proxy Function, Communications Hub Function,
6176 Type 1 Device or a Type 2 Device.

6177 **Device Based Access Control (DBAC)**

6178 Shall have the meaning defined in Section 13.7.3.

6179 **Device Certificate**

6180 Shall have the meaning set out in Section 12.

6181 **Device Log**

6182 Data item described in SMETS and CHTS.

6183 **Device Specifications**

6184 The document set comprising SMETS (including the IHDTs, HCalcSTS and PPMIDTs),
6185 and CHTS.

6186 **Digital Signature**

6187 The information appended to a Message which is created using the sender's Private Key,
6188 can be verified using the corresponding Public Key contained in the sender's Certificate, and
6189 provides the receiver with assurance that the sender is who they claim to be, the message
6190 has not been altered in transit and that the sender sent the Message.

6191	Digital Signing
6192	The creation of a Digital Signature.
6193	Digital Signing Certificate
6194	A Certificate which states that the Public Key contained within, and its associated Private
6195	Key, may be used for Digital Signing purposes.
6196	Distinguished Encoding Rules
6197	Shall have the meaning defined in http://www.itu.int/ITU-
6198	T/studygroups/com17/languages/X.690-0207.pdf
6199	DLMS COSEM
6200	Device Language Message Specification / Companion Specification for Energy Metering - an
6201	Application Layer protocol.
6202	Elliptic Curve DSA (ECDSA)
6203	The Elliptic Curve Digital Signature Algorithm (see
6204	http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf) as specified in Section 4.3.3.
6205	Encoding(X)
6206	The encoding of a variable length integer X, as specified in Section 3.3.
6207	Encryption
6208	The process of converting information in order to make it unintelligible other than to
6209	Authorised parties. Like terms shall be construed accordingly.
6210	Encryption Originator Counter
6211	Shall have the meaning defined in Table 23.
6212	Encryption Remote Party
6213	The Remote Party that encrypted Encrypted data items.
6214	Entity Identifier
6215	A 64 bit unsigned integer uniquely identifying a Smart Metering Entity.
6216	ESME
6217	Electricity Smart Metering Equipment, as described in the SMETS.
6218	Event
6219	A change in state generated by a Device in response to an internal or external trigger.
6220	Event Code
6221	A 16 bit unsigned integer taking the values specified in Section 16. The Alert Code and
6222	Event Code are the same for a given Event.
6223	Event Log
6224	Data item described in SMETS and CHTS.
6225	Execution Counter
6226	Shall have the meaning defined in Section 4.3.1.5.
6227	Firmware
6228	The embedded software programmes and / or data structures that control Devices.
6229	Firmware Distribution Receipt Alert

- 6230 Shall have the meaning set out in Section 11.2.6.
- 6231 **Force Replace**
- 6232 The means to instruct a Communications Hub to replace an ESME or GSME Firmware
6233 image that it holds, e.g. when the image has only been partially downloaded to the ESME or
6234 GSME. This enables recovery from failures.
- 6235 **Gas Proxy Function (GPF)**
- 6236 A Gas Proxy Function as defined in the Communications Hub Technical Specifications.
- 6237 **Galois Counter Mode (GCM)**
- 6238 The mode of operation specified in *NIST Special Publication 800-38D*.
- 6239 **GBZ**
- 6240 A set of structures in the GBCS which carry ZCL / ZSE commands.
- 6241 **General Block Transfer (GBT) / GBT Message**
- 6242 General Block Transfer is a DLMS COSEM mechanism for decomposing APDUs above
6243 maximum sizes that can be transported in to a number of smaller APDUs, which are no
6244 larger than the maximum sizes. A GBT Message is one of these smaller APDUs.
- 6245 **GMAC**
- 6246 Variant of GCM that is used to generate Message Authentication Code from non-
6247 Confidential data, as specified in *NIST Special Publication 800-38D*.
- 6248 **Green Book**
- 6249 The DLMS Green Book Version *DLMS UA 1000-2 Ed.8*. This document can be obtained
6250 from the DLMS User Association: <http://www.dlms.com>.
- 6251 **GSME**
- 6252 Gas Smart Metering Equipment, as described in the SMETS.
- 6253 **HAN Only Message**
- 6254 A Message where both the sender and recipient are Devices on the same Smart Metering
6255 Home Area Network.
- 6256 **HAN Connected Auxiliary Load Control Switch (HCALCS)**
- 6257 A Type 1 Device capable of communicating with an ESME.
- 6258 **Hashing**
- 6259 A repeatable process to create a fixed size condensed representation of a Message or any
6260 arbitrary data, as further set out in Section 4.3.3. Hash and like terms shall be construed
6261 accordingly.
- 6262 **HCALCS**
- 6263 HAN Connected ALCS.
- 6264 **HCALCSTS**
- 6265 The HAN Connected Auxiliary Load Control Switches (HCALCS) Technical Specification in
6266 SMETS.
- 6267 **Highest Prior Sequence Number**
- 6268 Shall have the meaning defined in Section 13.3.5.3.
- 6269 **Home Area Network Interface (HAN Interface)**

6270	A component of GSME, ESME, IHD or other Device that is capable of sending and receiving
6271	information to and from other Devices.
6272	IHDTs
6273	The In Home Display Technical Specifications in SMETS.
6274	IHD
6275	In Home Display.
6276	IHD Source Device
6277	An ESME or GPF.
6278	Initialization Vector (IV)
6279	An input to the Authenticated Encryption and Authenticated Decryption functions defined in
6280	<i>NIST Special Publication 800-38D</i> . Where the GBCS applies, it shall have the values as
6281	specified at Section 4.3.3.4.
6282	Inter-PAN
6283	Shall have the meaning defined in CHTS.
6284	Join
6285	The process of authorising two Devices to communicate at the application layer.
6286	Key
6287	Data used to determine the output of a cryptographic operation.
6288	Key Agreement
6289	A means to calculate a shared Key between two parties.
6290	Key Agreement Certificate
6291	A Certificate which states that the Public Key contained within, and its associated Private
6292	Key, may be used for Key Agreement purposes.
6293	Key Derivation Function (KDF)
6294	A function to generate derived keying material from a Shared Secret and other information.
6295	Known Remote Party (KRP)
6296	In the context of a specific Device, a Remote Party whose Security Credentials are stored on
6297	that Device in at least one Trust Anchor Cell.
6298	KRP Signature
6299	A Digital Signature generated by a Known Remote Party.
6300	Len(X)
6301	The number of octets in the variable length octet string X.
6302	MAC Header
6303	As defined in Section 6, a part of a message which is only present when the Message
6304	contains a MAC but which is additional to the MAC.
6305	Manufacturer Image Hash
6306	Shall have the meaning defined in Section 11.2.4.
6307	Mapping Table

- 6308 The spreadsheet detailing Use Cases and associated protocol requirements as embedded in
6309 Section 20.
- 6310 [Maximum Credit Threshold](#)
- 6311 Shall have the meaning defined in SMETS.
- 6312 [Maximum Meter Balance Threshold](#)
- 6313 Shall have the meaning defined in SMETS.
- 6314 [Message](#)
- 6315 A Command, Response or Alert.
- 6316 [Message Authentication](#)
- 6317 The process by which the receiver of a Message is provided with assurance that the sender
6318 is who they claim to be and that the Message is in the form originally sent.
- 6319 [Message Authentication Code \(MAC\)](#)
- 6320 The number incorporated in a Message to provide Message Authentication, as set out in
6321 Section 4.3.3.
- 6322 [Message Category](#)
- 6323 A grouping of Remote Party Messages.
- 6324 [Message Code](#)
- 6325 A 16 bit unsigned integer identifying the Use Case that the Message in question must
6326 conform to. Message Codes have the values specified in Section 15.
- 6327 [Message Identifier](#)
- 6328 Message Identifier shall be the concatenation of:
- 6329 • Business Originator ID;
- 6330 • Business Target ID;
- 6331 • CRA Flag; and
- 6332 • Originator Counter.
- 6333 [Message Series](#)
- 6334 Shall have the meaning defined in Section 7.2.11.1.
- 6335 [Message Template](#)
- 6336 A protocol-specific table defining the encoding of a Message.
- 6337 [Message Type](#)
- 6338 The Message Types are Command, Response or Alert.
- 6339 [Network Interface](#)
- 6340 A WAN Interface or HAN Interface.
- 6341 [Network Operator](#)
- 6342 In the context of a specific Device, the Known Remote Party whose Security Credentials are
6343 stored in the {networkOperator, digitalSignature, management} Trust Anchor
6344 Cell.
- 6345 [Object Identifier \(OID\)](#)

- 6346 An identifier used to name an object. Structurally, an OID consists of a node in a
6347 hierarchically-assigned namespace, formally defined using the ASN.1 standard.
- 6348 [Organisation Certificate](#)
- 6349 Shall have the meaning set out in Section 12.
- 6350 [Originator Counter](#)
- 6351 Shall have the meaning defined in Section 4.3.1.2.
- 6352 [OtherInfo](#)
- 6353 An input to the KDF with the meaning as specified in section 5.8.1 of *NIST Special*
6354 *Publication 800-56Ar2*.
- 6355 [Other User](#)
- 6356 A Remote Party which is not a Known Remote Party in relation to any Device, and so is
6357 always an Unknown Remote Party in any communication with a Device.
- 6358 [Outcome](#)
- 6359 The result of executing a Command, expressed as success or failure.
- 6360 [Payload](#)
- 6361 Part of the Message that provides the message-specific content.
- 6362 [Payment Mode](#)
- 6363 The information held on Smart Metering Equipment as described at sections 4 and 5 in
6364 SMETS.
- 6365 [Pending Private Key](#)
- 6366 A Private Key held on a Device for which a Device has not successfully received and
6367 processed a Device Certificate for the corresponding Public Key as defined in Section 13.5.
- 6368 [Personal Data](#)
- 6369 Any information comprising Personal Data as such term is defined in the Data Protection Act
6370 1998 at the date the GBCS is brought into force.
- 6371 [Plain Format](#)
- 6372 A Signature is a pair of integers, r and s. For the Elliptic Curve required by the GBCS, each
6373 can be represented as a 256 bit (or 32 octet) string. The Plain Format of a GBCS signature
6374 is the concatenation R || S where R is the 32 octet string representing r and S is the 32 octet
6375 string representing s. Thus, a GBCS Signature is an octet string of length 64.
- 6376 [Plaintext](#)
- 6377 An input to the Authenticated Encryption function and an output from the Authenticated
6378 Decryption function defined in *NIST Special Publication 800-38D*. Plaintext is the data
6379 whose Confidentiality is to be protected by Encryption. The encrypted form of the Plaintext is
6380 the Ciphertext.
- 6381 [PPMIDTS](#)
- 6382 The Prepayment Interface Device (PPMID) Technical Specification in SMETS.
- 6383 [Polyphase](#)
- 6384 ESME containing three measuring elements suitable for a polyphase supply with up to three
6385 phases and neutral.
- 6386 [Premise\(s\)](#)

- 6387 The premise(s) which is / are being Supplied.
- 6388 [Prepayment Daily Read Log](#)
- 6389 Shall have the meaning defined in SMETS.
- 6390 [Prepayment Interface Device \(PPMID\)](#)
- 6391 A Device that provides a User Interface for Prepayment Mode related information and
6392 Commands.
- 6393 [Prepayment Token Decimal \(PPTD\)](#)
- 6394 Shall have the meaning defined in Section 14.1.
- 6395 [Prepayment Top Up](#)
- 6396 The addition of credit to an ESME or GSME operating in prepayment mode.
- 6397 [Prepayment Top Up Token](#)
- 6398 Shall have the meaning defined in Section 14.1.
- 6399 [Private Digital Signing Key](#)
- 6400 A Private Key used for Digital Signing only.
- 6401 [Private Key](#)
- 6402 The key in a Public-Private Key Pair which must be kept secure by the entity to which it
6403 relates.
- 6404 [Private Key Cell](#)
- 6405 Shall have the meaning defined in Section 4.3.2.3. A Private Key Cell may be Current or
6406 Pending.
- 6407 [Private Key Agreement Key](#)
- 6408 A Private Key used for Key Agreement only.
- 6409 [Protection Against Replay](#)
- 6410 An attribute defined in a Use Case specifying whether a recipient Device is required to
6411 implement the Protection Against Replay mechanisms, as defined in Section 4.3.1.5, for the
6412 Command covered by the Use Case.
- 6413 [Protocol Data Unit \(PDU\)](#)
- 6414 Information delivered as a unit among peer entities of networks containing control
6415 information, address information or data.
- 6416 [Public Digital Signing Key](#)
- 6417 A Public Key used for Digital Signing only.
- 6418 [Public Key](#)
- 6419 The key in a Public-Private Key Pair which can be distributed to other parties.
- 6420 [Public Key Agreement Key](#)
- 6421 A Public Key used for Key Agreement only.
- 6422 [Public Key Security Credentials](#)
- 6423 Security Credentials which include a Public Key.
- 6424 [Public-Private Key Pair](#)
- 6425 Two mathematically related numbers that are used in Cryptographic Algorithms.

6426 [Recovery](#)

6427 In the context of a specific Device, the Known Remote Party whose Security Credentials are
 6428 stored in the {recovery, digitalSignature, management} Trust Anchor Cell.

6429 [Reliable Time](#)

6430 The state of the Device clock such that is within 10 seconds of UTC, synchronised with the
 6431 HAN time server and confirmed by Set Clock Command from the Remote Party whose
 6432 security Credentials are stored in the {supplier, digitalSignature, management}
 6433 Trust Anchor Cell.

6434 [Remote Party](#)

6435 An entity which is remote from the Premises and is able to either send Messages to or
 6436 receive Messages from a Device within the Premises, whether directly or via a third party.

6437 [Remote Party Alert](#)

6438 Shall have the meaning defined in Section 7.2.3.

6439 [Remote Party Command](#)

6440 Shall have the meaning defined in Section 7.2.1.

6441 [Remote Party Message](#)

6442 A Message where either the sender(s) or recipient(s) are not Devices.

6443 [Remote Party Role](#)

6444 A class of Remote Party in relation to which one or more Devices is capable of storing
 6445 Security Credentials.

6446 [Remote Party Role Code](#)

6447 An 8 bit unsigned integer which uniquely identifies a Remote Party Role. The value for each
 6448 Remote Party Role shall be as defined in Section 4.3.2.4.

6449 [Replay Attack](#)

6450 A form of attack on a Communications Link in which a valid information transmission is
 6451 repeated through interception and retransmission.

6452 [Response](#)

6453 A response to a Command received or sent over any interface.

6454 [Response Payload](#)

6455 The parts of a Response that are not related to Cryptographic Protections for integrity,
 6456 authenticity or non-repudiation, as defined in Section 7.2.2.

6457 [Role](#)

6458 The entitlement of a party to execute one or more Commands.

6459 [Root](#)

6460 In the context of a specific Device, the entity whose Security Credentials are stored in the
 6461 {root, keyCertSign, management} Trust Anchor Cell.

6462 [Secure Perimeter](#)

6463 A physical border surrounding ESME, GSME or the PPMID.

6464 [Security Credential Document](#)

6465 A Security Credential Document shall be defined as either a:

- 6466 • Device's Certificate; or a
- 6467 • Remote Party's Certificate; or a
- 6468 • Certification Authority Certificate
- 6469 [Security Credentials](#)
- 6470 Information used to identify and / or Authenticate a Device, Party or system.
- 6471 [Security Log](#)
- 6472 Data item described in SMETS and CHTS.
- 6473 [SHA-256](#)
- 6474 The Hashing algorithm of that name approved by the NIST (see
- 6475 http://csrc.nist.gov/groups/ST/toolkit/secure_hashing.html).
- 6476 [Shared Secret](#)
- 6477 A number which is established by two parties through the Key Agreement technique
- 6478 specified in this GBCS and which can be used as input to a KDF.
- 6479 [Shared Secret Key](#)
- 6480 A number which is derived using the KDF specified in this GBCS.
- 6481 [Smart Energy Code \(SEC\)](#)
- 6482 The document of that name, as designated by the Secretary of State under Condition 22 of
- 6483 the DCC Licence.
- 6484 [Smart Metering Device to Known Remote Party MAC \(SMD-KRP MAC\)](#)
- 6485 A MAC generated by a Device in relation to a Response or Alert which can only be verified
- 6486 by the Known Remote Party which is the target of the Response or Alert.
- 6487 [Smart Metering Entity](#)
- 6488 An entity that is either a Device or a Remote Party.
- 6489 [Smart Metering Equipment Technical Specifications \(SMETS\)](#)
- 6490 A version of the Smart Metering Equipment Technical Specifications set out in Schedule 9 of
- 6491 the Smart Energy Code.
- 6492 [Smart Metering Home Area Network \(SMHAN\)](#)
- 6493 The network enabling communications between the Devices recorded within a
- 6494 Communications Hubs' Device Log (as defined in CHTS).
- 6495 [SMD Signature](#)
- 6496 A Digital Signature generated by a Device.
- 6497 [Supplementary Originator Counter](#)
- 6498 Shall have the meaning defined in Section 23.
- 6499 [Supply](#)
- 6500 The supply of gas to Premises for GSME and the supply of electricity to Premises for ESME
- 6501 and 'Supplied' shall be construed accordingly.
- 6502 [Supplier](#)
- 6503 A person authorised by licence to Supply gas to Premises for GSME and a person
- 6504 authorised by licence to Supply electricity to Premises for ESME. In the context of a specific

- 6505 Device, the Known Remote Party whose Security Credentials are stored in the {supplier,
6506 digitalSignature, management} Trust Anchor Cell.
- 6507 **Tag**
- 6508 The first element within a Message Header or part of a Message that provides identification
6509 of the Message or part of Message that follows.
- 6510 **Tapping Off Mechanism (TOM)**
- 6511 Shall have the meaning defined in Section 10.3.4.
- 6512 **Tariff**
- 6513 The structure of prices and other charges relating to a Supply.
- 6514 **Tariff Block Counter Matrix**
- 6515 Data item described in SMETS.
- 6516 **TOU**
- 6517 Time of Use.
- 6518 **Transactional Atomicity**
- 6519 The type and order of the constituent parts of a Command.
- 6520 **Transitional Change of Supplier**
- 6521 In the context of a specific Device, the Known Remote Party whose Security Credentials are
6522 stored in relation to the Transitional Change of Supplier role.
- 6523 **Trust Anchor (TA)**
- 6524 A Trust Anchor represents a Remote Party via a Public Key and associated data stored on a
6525 Device. A Trust Anchor is used by the Device in specified cryptographic operations to
6526 determine whether it should act on Remote Party Commands received.
- 6527 **Trust Anchor Cell**
- 6528 A data store on a Device capable of storing one Trust Anchor. Each Trust Anchor Cell is for
6529 a fixed and pre-specified `keyUsage`, `cellUsage` and `remotePartyRole`.
- 6530 **Trust Anchor Management Protocol (TAMP)**
- 6531 A range of IETF RFCs relate to Trust Anchor Management, including:
- 6532 • [RFC4210] Adams, C., Farrell, S., Kause, T., and T. Mononen, 'Internet X.509 Public
6533 Key Infrastructure Certificate Management Protocol (CMP)', [RFC 4210](#), September
6534 2005.
 - 6535 • [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W.
6536 Polk, 'Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation
6537 List (CRL) Profile', [RFC 5280](#), May 2008.
 - 6538 • [RFC5914] Housley, R., Ashmore, S., and C. Wallace, 'Trust Anchor Format', [RFC](#)
6539 [5914](#), June 2010.
 - 6540 • [RFC5934] Housley, R., Ashmore, S., and C. Wallace, 'Trust Anchor Management
6541 Protocol (TAMP)', [RFC 5934](#), August 2010.
 - 6542 • [RFC6024] Reddy, R. and C. Wallace, 'Trust Anchor Management Requirements',
6543 [RFC 6024](#), October 2010.
- 6544 **Trusted Source**
- 6545 A source whose identity is confidently and reliably validated.

6546	Twin Element
6547	ESME containing two measuring elements.
6548	Type 1 Device
6549	A Device, other than GSME, ESME, Communications Hub, Communications Hub Function
6550	or Gas Proxy Function, that stores and uses the Security Credentials of other Devices for the
6551	purposes of communicating with them via its HAN Interface.
6552	Type 2 Device
6553	A Device that does not store or use the Security Credentials of other Devices for the
6554	purposes of communicating with them via its HAN Interface.
6555	Unauthorised
6556	Not Authorised.
6557	Unauthorised Physical Access
6558	Unauthorised access to the internal components of any Device within GSME or ESME
6559	through its Secure Perimeter.
6560	Unique Transaction Reference Number (UTRN)
6561	A 20 decimal digit number that is used to convey a Prepayment Top-Up Remote Party
6562	Command to an ESME / GSME.
6563	Unknown Remote Party (URP)
6564	In the context of a specific Device, a Remote Party whose Security Credentials are not
6565	stored on that Device.
6566	Upgrade Image
6567	Shall have the meaning defined in Section 11.2.2.
6568	Use Case
6569	The structure, format and processing of a Message.
6570	User Interface
6571	An interface for providing local human interaction with Devices which supports input and
6572	visual output.
6573	User Interface Command
6574	A Remote Party Command that is entered through the User Interface.
6575	UTC
6576	Coordinated Universal Time.
6577	UTRN Check Digit
6578	Shall have the meaning defined in Section 14.1.
6579	UTRN Counter Cache
6580	Shall have the meaning defined in Section 14.1.
6581	Variant Message
6582	A Message that does not fall in to any of the Message Categories defined in Section 6.
6583	Wide Area Network (WAN) Interface
6584	A component of a Communications Hub that is capable of sending and receiving information
6585	via the Wide Area Network Provider.

6586 **Wide Area Network (WAN) Provider**

6587 The organisation providing communications over the WAN Interface of the Communications
6588 Hub. Consequently, in the context of a specific Communications Hub, the Known Remote
6589 Party whose Security Credentials are stored in the {wanProvider, digitalSignature,
6590 management} Trust Anchor Cell.

6591 **ZigBee Cluster Library (ZCL)**

6592 The ZigBee Cluster Library Specification reference document as defined in the
6593 'Documentation Alignment' section of this GBCS.

6594 **ZigBee Smart Energy (ZSE) Profile (SEP)**

6595 The ZigBee Smart Energy (ZSE) Profile Specification 1.2a v1.0, reference 07-5356r19 as
6596 defined in the 'Documentation Alignment' section of this GBCS. Available from
6597 <http://zigbee.org/About/GBCSPartner.aspx>

22 Annex 1 – Additional DLMS Class

The class described below shall be supported by ESME. 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

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.

value_active

Provides information on the unit and the scalar for the value.

scal_unit_type: structure

{

 scalar,

	<pre> 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 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. </pre>
value_passive	<pre> 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 </pre>
scaler_unit_passive	<pre> 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 </pre>
activate_passive_value_time	

6602 22.2 Method description

Reset	<p>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.</p> <pre>data ::= integer(0)</pre>
activate_passive_value	<p>This method copies all attributes called ..._passive to the corresponding attributes called ..._active.</p> <pre>data ::= integer(0)</pre>

6603

23 Annex 2 – Counters and their use in transaction identification and Protection Against Replay – informative

Table 23 provides a summary of the Counters used in GB Smart Metering and outlines the purpose each serves in providing transaction identity, traceability and Protection Against Replay. References to the relevant sections of GBCS are included for each Counter where normative definitions can be found

Name	Description
Originator Counter	When combined with CRA Flag, Business Originator ID and Business Target ID the Originator Counter provides a unique Message Identifier. By uniquely identifying each Message, Devices can implement cryptographic requirements and measures to protect against Message replay. Originator Counters are always strictly numerically greater than any previous Originator Counter from a Message originator to the Message recipient (each of which may be a Remote Party or a Device). In certain cases the Originator Counter is used as an input to symmetric Key Derivation Functions. A specific numeric range of Originator Counter values are reserved for use in Prepayment Top Up Commands. See Section 4.3.1.2 for further detail
Execution Counter	The Execution Counter is stored by the Device for each Known Remote Party/Command combination where the Command requires protection against replay, except for Prepayment Top Up related Commands. Each Counter contains the last accepted Originator Counter value. Devices must discard Commands where the Originator Counter in the Command is not greater than the existing value of the Execution Counter stored on the Device for that Remote Party / Command combination. See Section 4.3.1.5 for further detail
Supplementary Originator Counter	The Supplementary Originator Counter is a Device generated number which is used to encrypt data returned in some Responses. The Supplementary Originator Counter is returned in the corresponding Response to allow the intended recipient to decrypt the encrypted data and validate the associated AE MAC. In line with Originator Counters, this Supplementary Originator Counter is always strictly numerically greater than any previous Supplementary Originator Counter or Originator Counter placed in previous messages by the Device). See Section 4.3.1.4 for further detail
Supplementary Remote Party Counter	GBCS does not specify the content of this field and the ACB can populate it with an identifier that is useful to the organisation on whose behalf the ACB sent the Command. See Section 4.3.1.4 for further detail
UTRN Counter	The UTRN Counter provides a specific Protection Against Replay mechanism for Prepayment Top Up. The UTRN Counter comprises the 32 most significant bits of the Originator Counter (this is a reserved range of Originator Counters where the least significant 32 bits are set to 0 for all Originator Counters used in relation to

	Prepayment Top Up Commands). If the UTRN Counter specified by a prepayment Command (whether entered locally or received over the WAN) is already in the UTRN Counter Cache on the Device or is less than the lowest value in the UTRN Counter Cache on the Device, then the Device will reject the UTRN. See Section 14 for further detail
PTUT Truncated Originator Counter	The PTUT Truncated Originator Counter is entered locally as part of a 20 digit decimal number. A Device then uses the algorithm detailed in Section 14.6 to derive the full 32 bit UTRN Counter for the Prepayment Top Up Command which is applied to a meter in Prepayment Mode. See Section 14 for further detail
Remote Party Floor Sequence Numbers	These Sequence Numbers are carried in the Update Security Credentials command and allow the replacement of Execution Counters and re-setting of the UTRN Counter Cache on the targeted Device. See Section 13.3.5.10 for further detail
Encryption Originator Counter	The Encryption Originator Counter is a key derivation input and is populated with the Originator Counter except where a Supplementary Originator Counter is included in a Response by a device. See Sections 4.3.1.4 and 8.3 for further detail

6610 Table 23: Counters and their use in transaction identification and Protection Against Replay

24 Annex 3 – ASN.1 modules – informative

This Annex collates all ASN.1 schema used in this GBCS. Please note that this is a duplicate; the authoritative content remains as documented in the appropriate section.

```
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
```

```
ActivateFirmware DEFINITIONS ::= BEGIN
```

```
    CommandPayload ::=
        SEQUENCE
        {
            -- specify the hash of the Manufacturer Image to be activated
```

```

6650     manufacturerImageHash                OCTET STRING,
6651
6652     -- the Originator Counter as in the Grouping Header of the Command
6653     originatorCounter                      INTEGER (0.. 18446744073709551615),
6654
6655     -- the date-time at which the Command is to execute, if future dated
6656     executionDateTime                      GeneralizedTime OPTIONAL
6657 }
6658
6659 ResponsePayload ::=
6660 {
6661     -- if the Command is future dated, the Response will not have any details of
6662     -- execution (those will be in the subsequent alert)
6663     commandAccepted                        NULL,
6664
6665     -- if the Command is for immediate execution, the Response will detail the
6666     -- outcomes
6667     executionOutcome                       ExecutionOutcome
6668 }
6669
6670 AlertPayload ::=
6671 {
6672     -- specify the Alert Code
6673     alertCode                             INTEGER(0..4294967295),
6674
6675     -- specify the date-time of execution
6676     executionDateTime                      GeneralizedTime,
6677
6678     -- the Originator Counter as in the Grouping Header of the corresponding Command
6679     originatorCounter                      INTEGER (0.. 18446744073709551615),
6680
6681     -- detail what happened when the future dated command was executed
6682     executionOutcome                       ExecutionOutcome
6683 }
6684
6685 ExecutionOutcome ::=
6686 {
6687     -- Specify whether the activation was successful or not
6688     activateImageResponseCode              ActivateImageResponseCode,
6689
6690     -- Specify the Device's now current firmware version. The value shall be four octets in length and shall correspond to the
6691     File Version field in the ZSE OTA Header structure.
6692     firmwareVersion                        OCTET STRING
6693 }

```



```

6694
6695 ActivateImageResponseCode ::= INTEGER
6696 {
6697     success                                (0),
6698     noImageHeld                            (1),
6699     hashMismatch                           (2),
6700     activationFailure                       (3)
6701 }
6702
6703 END

```

```

6704
6705 ProvideSecurityCredentialDetails DEFINITIONS ::= BEGIN
6706
6707 Command ::=
6708     SEQUENCE
6709     {
6710         -- Identify which of the Public Keys on the Device is to be used in verifying the Signature or MAC
6711         -- (so defining the nature of the verification by way of the KeyUsage parameter held on the
6712         -- Device for the Public Key so identified).
6713         authorisingRemotePartyTACellIdentifier    TrustAnchorCellIdentifier,
6714
6715         -- List the Remote Party Role(s) for which credential details are required
6716         remotePartyRolesCredentialsRequired      SEQUENCE OF RemotePartyRole
6717     }
6718
6719 Response ::=
6720     SEQUENCE OF RemotePartyDetails
6721
6722 RemotePartyDetails ::=
6723     SEQUENCE
6724     {
6725         -- Which Remote Party do these details relate to?
6726         remotePartyRole                          RemotePartyRole,
6727
6728         -- statusCode shall be success unless the role is not valid on this type of Device or there is a processing failure
6729         statusCode                               StatusCode,
6730
6731         -- What is the current Update Security Credentials Protection Against Replay number on the Device for this role, where there is
6732         -- such a number for this role?
6733         currentSeqNumber                         SeqNumber OPTIONAL,
6734
6735

```

```

6736
6737 -- What are the details held on the Device for each of the Cells related to this role? The list shall have between one and
6738 -- three entries (e.g. there will be one if role is transitional change of supplier; there may be three if role is supplier)
6739
6740 trustAnchorCellsDetails                               SEQUENCE OF TrustAnchorCellContents OPTIONAL
6741 }
6742
6743 SeqNumber ::=                                         INTEGER (0.. 18446744073709551615)
6744
6745 TrustAnchorCellContents ::=                           SEQUENCE
6746 {
6747 -- To what cryptographic use can the Public Key in this Cell be put? Some Remote Party Roles
6748 -- (e.g. supplier) can have more than one Public Key on a Device and each one would only have
6749 -- a single cryptographic use.
6750
6751 trustAnchorCellKeyUsage                               KeyUsage,
6752
6753 -- trustAnchorCellUsage is to allow for multiple Public Keys of the same keyUsage for the same Remote
6754 -- Party Role. This will be absent except where used to refer to the Supplier Key Agreement Key.
6755 -- This Key is used solely in relation to validating Supplier generated MACs on Prepayment Top Up transactions.
6756
6757 trustAnchorCellUsage                                 CellUsage DEFAULT management,
6758
6759 -- The subjectUniqueID which shall be the 64 bit Entity Identifier of the Security Credentials in this Trust Anchor Cell.
6760
6761 existingSubjectUniqueID                               OCTET STRING,
6762
6763 -- The APKI requirements mean that KeyIdentifier attributes will all be 8 byte SHA-1 Hashes.
6764 -- existingSubjectKeyIdentifier shall be set accordingly based on the contents of the Trust Anchor Cell
6765
6766 existingSubjectKeyIdentifier                         OCTET STRING
6767 }
6768
6769 TrustAnchorCellIdentifier ::=                         SEQUENCE
6770 {
6771 -- Which Remote Party Role does this Cell relate to?
6772
6773 trustAnchorCellRemotePartyRole                       RemotePartyRole,
6774
6775 -- To what cryptographic use can the Public Key in this Cell be put? Some Remote Party Roles
6776 -- (e.g. supplier) can have more than one Public Key on a Device and each one would only have
6777 -- a single cryptographic use.
6778
6779 trustAnchorCellKeyUsage                               KeyUsage,

```

```

6780
6781 -- trustAnchorCellUsage is to allow for multiple Public Keys of the same keyUsage for the same Remote
6782 -- Party Role. This may be absent except where use to refer to the Supplier Key
6783 -- Agreement Key used solely in relation to validating Supplier generated MACs on Prepayment Top Up transactions
6784
6785 trustAnchorCellUsage                                CellUsage DEFAULT management
6786 }
6787
6788 CellUsage ::=                                     INTEGER {management(0), prePaymentTopUp(1)}
6789
6790 RemotePartyRole ::=                               INTEGER
6791 {
6792 -- Define the full set of Remote Party Roles in relation to which a Device may need to undertake
6793 -- processing. Note that most Devices will only support processing in relation to a subset of these.
6794
6795 root                                                (0),
6796 recovery                                            (1),
6797 supplier                                            (2),
6798 networkOperator                                    (3),
6799 accessControlBroker                                (4),
6800 transitionalCoS                                    (5),
6801 wanProvider                                        (6),
6802 issuingAuthority                                  (7), -- Devices will receive such Certificates but they do not
6803 -- need to store them over an extended period
6804
6805
6806 -- The 'other' RemotePartyRole is for a party whose role does not allow it to invoke any Device function apart from
6807 -- UpdateSecurityCredentials. This is to allow for Device functionality to be locked out of usage until a valid
6808 -- Remote Party can be identified e.g. where roles cannot be fixed until a Device is bought in to operation
6809 other                                              (127)
6810 }
6811
6812
6813 -- KeyUsage is only repeated here for ease of reference. It is defined in RFC 5912
6814
6815 KeyUsage ::=                                       BIT STRING
6816 {
6817 -- Define valid uses of Public Keys.
6818
6819 digitalSignature                                    (0),
6820 contentCommitment                                  (1), -- not valid for GBCS compliant transactions
6821 keyEncipherment                                    (2), -- not valid for GBCS compliant transactions
6822 dataEncipherment                                   (3),

```

```

6824 keyAgreement                (4),
6825 keyCertSign                 (5),
6826 cRLSign                     (6),
6827 encipherOnly                 (7),
6828 decipherOnly                 (8)    -- not valid for GBCS compliant transactions
6829 }
6830
6831 -- The GBCS only allows for a constrained set of Trust Anchor Cell operations and so the list of possible outcomes
6832 -- is more limited than in IETF RFC 5934. The list below is that more constrained subset
6833
6834 StatusCode ::=                ENUMERATED {
6835
6836 success                        (0),
6837
6838 -- trustAnchorNotFound indicates that details of a trust anchor were requested, but the referenced trust anchor
6839 -- is not represented on the Device
6840
6841 trustAnchorNotFound           (25),
6842
6843 other                         (127)}
6844
6845
6846 END

```

```

6847
6848 UpdateSecurityCredentials DEFINITIONS ::= BEGIN
6849
6850 CommandPayload ::= SEQUENCE
6851 {
6852 -- Provide details to allow the Device to identify the Remote Party Role authorising
6853 -- this Command, check whether the rest of the payload is allowable, prevent replay attacks
6854 -- and allow counters / counter caches on the Device to be reset, if the Command changes the Remote Party
6855 -- in control.
6856 -- The Remote Party authorising the Command is that party which generated the KRP Signature (or the Access Control Broker
6857 -- if there is no KRP Signature)
6858
6859 authorisingRemotePartyControl    AuthorisingRemotePartyControl,
6860
6861 -- One TrustAnchorReplacement structure is required for each Trust Anchor Cell that is to be updated
6862
6863 replacements                     SEQUENCE OF TrustAnchorReplacement,
6864
6865 -- Provide the certificates needed to undertake Certification Path Validation of the new

```

```

6866 -- end entity certificate against the root public key held on the Device. The number of these may be less
6867 -- than the number of replacement certificates (e.g. a supplier may replace all of its certificates but
6868 -- may only need to supply one Certification Authority Certificate to link them all back to the root public
6869 -- key as currently stored on the Device.
6870
6871 certificationPathCertificates          SEQUENCE OF Certificate,
6872
6873 -- If the Command is to be future dated, specify the date-time at which the certificate replacement is to happen
6874
6875 executionDateTime                      GeneralizedTime OPTIONAL
6876
6877 }
6878
6879 ResponsePayload ::=                   SEQUENCE
6880 {
6881     -- if the Command is future dated, the Response will not have any details of execution (those will be in the subsequent alert)
6882
6883     commandAccepted                    NULL,
6884
6885     -- if the Command is for immediate execution, the Response will detail the outcomes
6886
6887     executionOutcome                   ExecutionOutcome OPTIONAL
6888
6889 }
6890
6891 AlertPayload ::=                      SEQUENCE
6892 {
6893     -- specify the Alert Code
6894     alertCode                          INTEGER(0..4294967295),
6895
6896     -- specify the date-time of execution
6897     executionDateTime                  GeneralizedTime,
6898
6899     -- detail what happened when the future dated Command was executed
6900
6901     executionOutcome                   ExecutionOutcome
6902
6903 }
6904
6905 ExecutionOutcome ::=                  SEQUENCE
6906 {
6907     -- Provide details of the corresponding Command that may not be in the standard GBCS message header. Specifically the
6908     -- mode in which the Command was invoked, the Originator Counter in the original Command and the resulting changes to any
6909

```

```

6910 -- replay counters held on the Device
6911
6912 authorisingRemotePartySeqNumber      SeqNumber,
6913 credentialsReplacementMode          CredentialsReplacementMode,
6914 remotePartySeqNumberChanges          SEQUENCE OF RemotePartySeqNumberChange,
6915
6916 -- For each replacement in the Command, detail the outcome and impacted parties
6917
6918 replacementOutcomes                  SEQUENCE OF ReplacementOutcome
6919
6920 }
6921
6922 AuthorisingRemotePartyControl ::=      SEQUENCE
6923 {
6924 -- Specify the replacement mode so that the Device can check that the Remote Party Role is allowed to
6925 -- authorise this type of replacement and that all replacements in the payload are allowed within this
6926 -- replacement mode
6927
6928 credentialsReplacementMode            CredentialsReplacementMode,
6929
6930 -- Only if credentialsReplacementMode = anyByContingency, provide the symmetric key to decrypt
6931 -- the Contingency Public Key in the (root, digitalSignature, management) Trust Anchor Cell
6932
6933 plaintextSymmetricKey                 [0] IMPLICIT OCTET STRING OPTIONAL,
6934
6935 -- Specify whether the time based checks as part of any Certificate Path Validation should be applied
6936
6937 applyTimeBasedCPVChecks                [1] IMPLICIT INTEGER {apply(0), disapply(1)} DEFAULT apply,
6938
6939 -- Identify which of the Public Keys on the Device is to be used in checking KRP Signature
6940 -- 'authorisingRemotePartyTACellIdentifier' may only be omitted when
6941 -- the access control broker is updating its own credentials and the target device is not a CHF.
6942 -- In all other cases it is mandatory.
6943
6944 authorisingRemotePartyTACellIdentifier [2] IMPLICIT TrustAnchorCellIdentifier OPTIONAL,
6945
6946 -- Specify the Originator Counter for the Remote Party Applying KRP Signature, or (for the
6947 -- Access Control Broker changing its credentials) the Access Control Broker's Originator Counter.
6948
6949 authorisingRemotePartySeqNumber        [3] IMPLICIT SeqNumber,
6950
6951 -- If the Command is to effect a change of control, then newTrustAnchorFloorSeqNumber must be included
6952 -- and will be the value used to prevent replay of Update Security Credentials Commands for the
6953 -- new controlling Remote Party.

```

```

6954
6955 newRemotePartyFloorSeqNumber          [4] IMPLICIT SeqNumber OPTIONAL,
6956
6957 -- Some Commands on the Device may use a different Originator Counter sequence for Protection Against Replay. At this
6958 -- version of the GBCS, the only example is the Prepayment Top Up Command on ESME and GSME. The
6959 -- SpecialistSeqNumber structure allows such Counters to also be reset on change of control.
6960
6961 newRemotePartySpecialistFloorSeqNumber  [5] IMPLICIT SEQUENCE OF SpecialistSeqNumber OPTIONAL,
6962
6963 -- In some cases, one party acting in one Remote Party Role may be replacing certificates for a different Remote Party Role.
6964 -- In some cases, sequence counters need also to be reset for those other Remote Party Role(s)
6965
6966 otherRemotePartySeqNumberChanges        [6] IMPLICIT SEQUENCE OF RemotePartySeqNumberChange OPTIONAL
6967 }
6968
6969 RemotePartySeqNumberChange ::=
6970 {
6971   otherRemotePartyRole                RemotePartyRole,
6972   otherRemotePartyFloorSeqNumber       SeqNumber,
6973   newRemotePartySpecialistFloorSeqNumber SEQUENCE OF SpecialistSeqNumber OPTIONAL
6974 }
6975
6976 SpecialistSeqNumber ::=
6977 {
6978   -- Specify the usage of the SeqNumber
6979   seqNumberUsage                      SeqNumberUsage,
6980
6981   -- Specify the associated SeqNumber
6982   seqNumber                          SeqNumber
6983 }
6984
6985 SeqNumberUsage ::=
6986 {
6987   -- Define the full set of discrete usages on a Device. The only specialist
6988   -- counter is for Prepayment Top Up (which is set independently of other counters). This may only be
6989   -- included when changing Supplier Security Credentials on an ESME or GSME.
6990
6991   prepaymentTopUp                     (0)
6992 }
6993
6994 SeqNumber ::=
6995
6996
6997 TrustAnchorReplacement ::=
6998
6999
7000

```

```

6998 {
6999 -- Provide the new end entity certificate
7000
7001 replacementCertificate          Certificate,
7002
7003 -- Specify where it is to go (specifically which Trust Anchor Cell is to have its details replaced using
7004 -- the new end entity certificate)
7005
7006 targetTrustAnchorCell          TrustAnchorCellIdentifier
7007 }
7008
7009
7010 ReplacementOutcome ::=          SEQUENCE
7011 {
7012   affectedTrustAnchorCell      TrustAnchorCellIdentifier,
7013   statusCode                   StatusCode,
7014
7015   -- The GBCS Certificate requirements mean that the subjectUniqueID attribute in the subject field of a certificate will always
7016   -- contain the 64 bit unique number that equates to Entity Identifier. existingSubjectUniqueID should be set
7017   -- accordingly based on the contents of the Trust Anchor Cell prior to Command processing.
7018
7019   existingSubjectUniqueID      OCTET STRING,
7020
7021   -- The GBCS Certificate requirements mean that subjectKeyIdentifier attributes will all be 8 byte SHA-1 Hashes.
7022   -- existingSubjectKeyIdentifier should be set accordingly based on the contents of the Trust Anchor Cell prior to
7023   -- Command processing.
7024
7025   existingSubjectKeyIdentifier  OCTET STRING,
7026
7027   -- The subjectUniqueID in the subject field of the certificate in this TrustAnchorReplacement
7028
7029   replacingSubjectUniqueID      OCTET STRING,
7030
7031   -- The subjectKeyIdentifier in the certificate in this TrustAnchorReplacement
7032
7033   replacingSubjectKeyIdentifier OCTET STRING
7034 }
7035
7036 TrustAnchorCellIdentifier ::=    SEQUENCE
7037 {
7038   -- Which Remote Party Role does this Cell relate to?
7039
7040   trustAnchorCellRemotePartyRole RemotePartyRole,
7041

```



```

7042 -- To what cryptographic use can the Public Key in this Cell be put? Some Remote Party Roles
7043 -- (e.g. supplier) can have more than one Public Key on a Device and each one would only have
7044 -- a single cryptographic use.
7045
7046 trustAnchorCellKeyUsage                                KeyUsage,
7047
7048 -- trustAnchorCellUsage is to allow for multiple Public Keys of the same keyUsage for the same Remote
7049 -- Party Role. It will be absent except where used to refer to the Supplier Key
7050 -- Agreement Key used solely in relation to validating Supplier generated MACs on Prepayment Top Up
7051 -- transactions
7052
7053 trustAnchorCellUsage                                CellUsage DEFAULT management
7054 }
7055
7056 CellUsage ::=                                         INTEGER {management(0), prePaymentTopUp(1)}
7057
7058 RemotePartyRole ::=                                   INTEGER
7059 {
7060 -- Define the full set of Remote Party Roles in relation to which a Device may need to undertake
7061 -- processing. Note that most Devices will only support a subset of these.
7062
7063 root                                                  (0),
7064 recovery                                              (1),
7065 supplier                                              (2),
7066 networkOperator                                      (3),
7067 accessControlBroker                                  (4),
7068 transitionalCoS                                       (5),
7069 wanProvider                                           (6),
7070 issuingAuthority                                     (7),    -- Devices will receive such Certificates but they do not need to store
7071                                           -- them over an extended period
7072
7073 -- The 'other' RemotePartyRole is for a party whose role does not allow it to invoke any Device function apart from
7074 -- UpdateSecurityCredentials. This is to allow for Device functionality to be locked out of usage until a valid
7075 -- Remote Party can be identified e.g. where roles cannot be fixed until a Device is brought in to operation
7076
7077 other                                                  (127)
7078
7079 }
7080
7081 -- KeyUsage is only repeated here for clarity. It is defined in RFC 5912
7082
7083 KeyUsage ::=                                         BIT STRING
7084 {
7085 -- Define valid uses of Public Keys held by Devices in their Trust Anchor Cells.

```

```

7086
7087 digitalSignature          (0),
7088 contentCommitment        (1),  -- not valid for GBCS compliant transactions
7089 keyEncipherment          (2),  -- not valid for GBCS compliant transactions
7090 dataEncipherment         (3),  -- not valid for GBCS compliant transactions
7091 keyAgreement             (4),
7092 keyCertSign              (5),
7093 cRLSign                  (6),
7094 encipherOnly             (7),  -- not valid for GBCS compliant transactions
7095 decipherOnly            (8)  -- not valid for GBCS compliant transactions
7096 }
7097
7098 CredentialsReplacementMode ::=          INTEGER
7099 {
7100 -- Define the valid combinations as to which Remote Party Roles can replace which kinds of Trust Anchors.
7101
7102 -- Normal operational replacement modes
7103 rootBySupplier            (0),
7104 rootByWanProvider        (1),
7105 supplierBySupplier        (2),
7106 networkOperatorByNetworkOperator (3),
7107 accessControlBrokerByACB  (4),
7108 wanProviderByWanProvider  (5),
7109 transCoSByTransCoS       (6),
7110 supplierByTransCoS       (7),
7111
7112 -- Recovery modes
7113 anyExceptAbnormalRootByRecovery (8),
7114 anyByContingency            (9)
7115 }
7116
7117 -- The GBCS only allows for a constrained set of Trust Anchor Cell operations and so the list of possible outcomes
7118 -- is more limited than in RFC 5934. The list below is that more constrained subset
7119
7120 StatusCode ::=          ENUMERATED {
7121
7122 success                (0),
7123
7124 -- badCertificate is used to indicate that the syntax for one or more certificates is invalid.
7125
7126 badCertificate          (5),
7127
7128 -- noTrustAnchor is used to indicate that the authorityKeyIdentifier does not identify the public key of a
7129 -- trust anchor or a certification path that terminates with an installed trust anchor

```

```

7130
7131 noTrustAnchor                                (10),
7132
7133 -- insufficientMemory indicates that the update could not be processed because the Device did not
7134 -- have sufficient memory
7135
7136 insufficientMemory                            (17),
7137
7138 -- contingencyPublicKeyDecrypt indicates that the update could not be processed because an error occurred while
7139 -- decrypting the contingency public key.
7140
7141 contingencyPublicKeyDecrypt                    (22),
7142
7143 -- trustAnchorNotFound indicates that a change to a trust anchor was requested, but the referenced trust anchor
7144 -- is not represented in the Trust Anchor Cell.
7145
7146 trustAnchorNotFound                          (25),
7147
7148 -- resourcesBusy indicates that the resources necessary to process the replacement are not available at the
7149 -- present time, but the resources might be available at some point in the future.
7150
7151 resourcesBusy                                (30),
7152
7153 -- other indicates that the update could not be processed, but the reason is not covered by any of the assigned
7154 -- status codes. Use of this status code SHOULD be avoided.
7155
7156 other                                         (127) }
7157
7158 END

```

```

7159
7160 IssueSecurityCredentials DEFINITIONS ::= BEGIN
7161
7162 CommandPayload ::=                               SEQUENCE
7163 {
7164     -- specify the keyUsage to which the generated key-pair will be put, if subsequently authorised
7165     keyUsage                               KeyUsage
7166 }
7167
7168 ResponsePayload ::=                               CHOICE
7169
7170 {
7171

```

```

7172      -- if the Command was successful, provide the generated Certification Request. CertificationRequest shall
7173      -- be as defined in ASN.1 by IETF RFC 5912. For reference, it is in the section headed 'ASN.1 Module for RFC 2986'
7174      certificationRequest          CertificationRequest,
7175
7176      -- if the Command was unsuccessful, detail the failure
7177
7178      issueCredentialsResponseCode    IssueCredentialsResponseCode
7179  }
7180
7181  -- KeyUsage is only repeated here for ease of reference. It is defined in IETF RFC 5912
7182
7183  KeyUsage ::=
7184  {
7185      -- Define valid uses of Public Keys held by Devices in their Trust Anchor Cells.
7186
7187      digitalSignature                (0),
7188      contentCommitment               (1),    -- not valid for GBCS compliant transactions
7189      keyEncipherment                (2),    -- not valid for GBCS compliant transactions
7190      dataEncipherment               (3),    -- not valid for GBCS compliant transactions
7191      keyAgreement                   (4),
7192      keyCertSign                    (5),    -- not valid for this Use Case
7193      cRLSign                        (6),    -- not valid for this Use Case
7194      encipherOnly                   (7),    -- not valid for GBCS compliant transactions
7195      decipherOnly                   (8)     -- not valid for GBCS compliant transactions
7196  }
7197
7198  IssueCredentialsResponseCode ::=
7199  {
7200      invalidKeyUsage                 (1),
7201      keyPairGenerationFailed         (2),
7202      cRProductionFailed              (3)
7203  }
7204
7205  END
7206
7207
7208  UpdateDeviceCertificateonDevice DEFINITIONS ::= BEGIN
7209
7210  CommandPayload ::=
7211      -- provide the certificate which the Device is to store
7212      -- the ASN.1 specification of certificate shall be as defined in IETF RFC 5912 for IETF RFC 5280
7213

```

```

7214 ResponsePayload ::=                                UpdateDeviceCertResponseCode
7215
7216     -- if the Command was unsuccessful, detail the failure; otherwise respond with success
7217
7218 UpdateDeviceCertResponseCode ::=                    INTEGER
7219 {
7220     success                                           (0),
7221     invalidCertificate                               (1),
7222     wrongDeviceIdentity                             (2),
7223     invalidKeyUsage                                  (3),
7224     noCorrespondingKeyPairGenerated                  (4),
7225     wrongPublicKey                                   (5),
7226     certificateStorageFailed                         (6),
7227     privateKeyChangeFailed                          (7)
7228 }
7229
7230 END

```

```

7231
7232 ProvideDeviceCertificateFromDevice DEFINITIONS ::= BEGIN
7233
7234 CommandPayload ::=                                SEQUENCE
7235 {
7236     -- specify the KeyUsage of the Certificate to be returned
7237     keyUsage                                         KeyUsage
7238 }
7239
7240 ResponsePayload ::=                                CHOICE
7241 {
7242     -- if the Command was successful, provide the certificate
7243     certificate                                     Certificate,
7244     -- if the Command was unsuccessful, detail the failure
7245     provideDeviceCertResponseCode                   ProvideDeviceCertResponseCode
7246 }
7247
7248
7249 -- KeyUsage is only repeated here for ease of reference. It is defined in RFC 5912
7250
7251
7252 KeyUsage ::=                                       BIT STRING

```

```

7256 {
7257 -- Define valid uses of Public Keys held by Devices in their Trust Anchor Cells.
7258
7259 digitalSignature          (0),
7260 contentCommitment        (1),    -- not valid for GBCS compliant transactions
7261 keyEncipherment          (2),    -- not valid for GBCS compliant transactions
7262 dataEncipherment         (3),    -- not valid for GBCS compliant transactions
7263 keyAgreement             (4),
7264 keyCertSign              (5),    -- not valid for this Use Case
7265 cRLSign                  (6),    -- not valid for this Use Case
7266 encipherOnly             (7),    -- not valid for GBCS compliant transactions
7267 decipherOnly             (8)    -- not valid for GBCS compliant transactions
7268 }
7269
7270 ProvideDeviceCertResponseCode ::= INTEGER
7271 {
7272     invalidKeyUsage        (1),
7273     noCertificateHeld      (2),
7274     certificateRetrievalFailure (3)
7275 }
7276
7277
7278 END

```

```

7279
7280 JoinDevice DEFINITIONS ::= BEGIN
7281
7282 CommandPayload ::= SEQUENCE
7283 {
7284     -- specify which type of joining is being authorised and,
7285     -- for Method A Joins, the role the Device is to play
7286
7287     joinMethodAndRole      JoinMethodAndRole,
7288
7289     -- specify the Entity Identifier of the Device which is to be Joined with
7290
7291     otherDeviceEntityIdentifier OCTET STRING,
7292
7293     -- specify the DeviceType of that other Device
7294
7295     otherDeviceType         DeviceType,
7296
7297     -- provide the other Device's Key Agreement certificate, if and only if this

```

```

7298      -- is a join between a gSME and a type1PrepaymentInterfaceDevice.
7299      -- Certificate shall be as defined in IETF RFC 5912
7300
7301      otherDeviceCertificate                      Certificate OPTIONAL
7302
7303  }
7304
7305  -- detail whether the Command successful executed or, if it didn't,
7306  -- what the failure reason was
7307
7308  ResponsePayload ::=
7309
7310      JoinResponseCode
7311
7312  JoinMethodAndRole ::=
7313  {
7314      -- methodB is to be used where the other Device is a Type 2 Device or GPF.
7315      -- methodC is used where the Devices involved are a GSME and a PPMID.
7316      -- methodA is used otherwise.
7317      -- methodAInitiator is used where the Device this Command is targeted at
7318      -- should initiate the Key Agreement process
7319      -- methodAResponder is used where the Device this Command is targeted at
7320      -- should respond in the Key Agreement process, but shall not initiate it
7321
7322      methodAInitiator                      (0),
7323      methodAResponder                     (1),
7324      methodB                             (2),
7325      methodC                             (3)
7326  }
7327
7328  DeviceType ::=
7329  {
7330      gSME                                (0),
7331      eSME                                (1),
7332      communicationsHubCommunicationsHubFunction (2),
7333      communicationsHubGasProxyFunction (3),
7334      type1HANConnectedAuxiliaryLoadControlSwitch (4),
7335      type1PrepaymentInterfaceDevice (5),
7336      type2                                (6)
7337  }
7338
7339  JoinResponseCode ::=
7340  {
7341      success                                (0),
7342      invalidMessageCodeForJoinMethodAndRole (1),
7343      invalidJoinMethodAndRole               (2),

```

```

7342     incompatibleWithExistingEntry           (3),
7343     deviceLogFull                             (4),
7344     writeFailure                             (5),
7345     keyAgreementNoResources                   (6),
7346     keyAgreementUnknownIssuer                 (7),
7347     keyAgreementUnsupportedSuite               (8),
7348     keyAgreementBadMessage                    (9),
7349     keyAgreementBadKeyConfirm                 (10),
7350     invalidOrMissingCertificate                (11)
7351 }
7352
7353 END

```

```

7354
7355 UnjoinDevice DEFINITIONS ::= BEGIN
7356
7357     CommandPayload ::=                               OtherDeviceEntityIdentifier
7358         -- specify the Entity Identifier of the Device for which authorisation
7359         -- is to be removed
7360
7361         OtherDeviceEntityIdentifier ::=               OCTET STRING
7362
7363     ResponsePayload ::=                               UnjoinResponseCode
7364
7365         -- detail whether the Command successful executed or, if it didn't,
7366         -- what the failure reason was
7367
7368     UnjoinResponseCode ::=                           INTEGER
7369     {
7370         success                                     (0),
7371         otherDeviceNotInDeviceLog                    (1),
7372         otherFailure                                 (2)
7373     }
7374
7375 END

```

```

7376
7377 ReadDeviceLog DEFINITIONS ::= BEGIN
7378
7379     CommandPayload ::=                               NULL
7380
7381     ResponsePayload ::=                               SEQUENCE
7382     {

```



```

7383      -- detail whether the Command successful
7384
7385      readLogResponseCode          ReadLogResponseCode,
7386
7387      -- if it was, return the Log Entries
7388      deviceLogEntries             SEQUENCE OF DeviceLogEntry OPTIONAL
7389  }
7390
7391  DeviceLogEntry ::=
7392  {
7393      deviceIdentifier             OCTET STRING,
7394      deviceType                   DeviceType
7395  }
7396
7397  DeviceType ::=
7398  {
7399      gSME                         (0),
7400      eSME                         (1),
7401      communicationsHubCommunicationsHubFunction (2),
7402      communicationsHubGasProxyFunction (3),
7403      type1HANConnectedAuxiliaryLoadControlSwitch (4),
7404      type1PrepaymentInterfaceDevice (5),
7405      type2                         (6)
7406  }
7407
7408
7409  ReadLogResponseCode ::=
7410  {
7411      success                       (0),
7412      readFailure                   (1)
7413  }
7414
7415  END

```

```

7416
7417  GPFDDeviceLog DEFINITIONS ::= BEGIN
7418
7419  BackupAlertPayload ::=
7420  {
7421      -- specify the Alert Code
7422      alertCode                     INTEGER(0..4294967295),
7423
7424      -- specify the date-time of the backup

```

```

7425         backupDateTime                               GeneralizedTime,
7426
7427         -- detail the entries in the Device Log now that the change has been made
7428         deviceLogEntries                               SEQUENCE OF DeviceLogEntry
7429
7430     }
7431
7432     RestoreCommandPayload ::=                               SEQUENCE
7433     {
7434         -- list the Device Log entries that are to be added
7435         deviceLogEntries                               SEQUENCE OF DeviceLogEntry
7436
7437     }
7438
7439     DeviceLogEntry ::=                               SEQUENCE
7440     {
7441         -- specify the Entity Identifier of the Device
7442         deviceEntityIdentifier                         OCTET STRING,
7443
7444         -- specify the DeviceType of that Device
7445
7446         deviceType                                     DeviceType
7447     }
7448
7449     RestoreResponsePayload ::=                               SEQUENCE
7450     {
7451         -- for each DeviceLog Entry, detail whether the Command successfully executed or, if it didn't, what the failure reason was
7452
7453         restoreOutcomes                               SEQUENCE OF RestoreOutcome
7454
7455     }
7456
7457     RestoreOutcome ::=                               SEQUENCE
7458     {
7459         deviceLogEntry                               DeviceLogEntry,
7460         joinResponseCode                             JoinResponseCode
7461     }
7462
7463     DeviceType ::=                               INTEGER
7464     {
7465         gSME                                           (0),
7466         eSME                                           (1),
7467         communicationsHubCommunicationsHubFunction    (2),

```

```
7469     communicationsHubGasProxyFunction      (3),
7470     type1HANConnectedAuxiliaryLoadControlSwitch (4),
7471     type1PrepaymentInterfaceDevice          (5),
7472     type2                                    (6)
7473 }
7474
7475 JoinResponseCode ::= INTEGER
7476 {
7477     success                                (0),
7478     invalidMessageCodeForJoinMethodAndRole (1),
7479     invalidJoinMethodAndRole              (2),
7480     incompatibleWithExistingEntry         (3),
7481     deviceLogFull                        (4),
7482     writeFailure                         (5),
7483     keyAgreementNoResources               (6),
7484     keyAgreementUnknownIssuer             (7),
7485     keyAgreementUnsupportedSuite          (8),
7486     keyAgreementBadMessage                (9),
7487     keyAgreementBadKeyConfirm             (10),
7488     invalidOrMissingCertificate           (11)
7489 }
7490
7491
7492 END
7493
```

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.

- 7536 GBCS lays out specific requirements regarding the use of Inter-PAN communications
7537 between a Communications Hub and an HHT. These are stated in Section 10.
- 7538 GBCS bars or modifies some internal Device behaviour that is specified in ZigBee
7539 standards, for example:
- 7540 • The ZSE capability for the OTA Client to self-activate any Firmware is not allowed, as
7541 stated in Section 11.2.1.
 - 7542 • The ZSE constraint requiring Trust Center involvement in CBKE shall not be applied,
7543 as stated in Section 13.7.2.
- 7544 GBCS does not require ZSE certification either for non-standard ZSE features or for non-
7545 ZSE features in GBCS.

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

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).

7586 For Devices other than ESME and CHF, the GBCS requires support for equivalents of the
 7587 Green Book's access and data notification services, but uses GBZ or ASN.1 specific
 7588 structures.

7589 For all Devices, the GBCS requires that the management of X.509 certificates and Device's
 7590 key pairs is undertaken using ASN.1 messages derived from the IETF's TAMP RFCs.

7591 Over the HAN, the GBCS mandates, for all Devices, the use of ZigBee for the
 7592 communication layers below the DLMS/COSEM Application Layer and so does not allow the
 7593 use of the equivalent Green Book communication profiles. (WAN transport is outside GBCS
 7594 scope).

7595 For ESME and GSME, distribution of firmware is through the ZSE OTA mechanism.

7596 **26.3.2 Specifies an approach that is derived from DLMS COSEM but**
 7597 **the approach is not part of the specification**

7598 GBCS specifies the use of a Class 9000 object, not specified in the Blue Book.

7599 For large Remote Party Messages, the GBCS uses DLMS type structures but in a way not
 7600 specified in the DLMS COSEM specification.

7601 Messages using pairwise key agreement between GSME and PPMID uses a structure
 7602 similar to DLMS COSEM's message structure, but that structure is not part of the DLMS
 7603 COSEM specification.

7604 **26.3.3 Specifies an approach that uses parts of the DLMS COSEM**
 7605 **but varies from it on specific points**

7606 For all bar Type 2 Devices, the DLMS general-signing structure is used in all Remote Party
 7607 Messages but the signature field is not populated in Messages that do not require a
 7608 signature (i.e. those that are not critical).

7609 For all bar Type 2 Devices, the GBCS uses the general-ciphering structure for Remote Party
 7610 Messages that require a MAC. The GBCS leaves most values empty in the header part of
 7611 the structure (these values are either in the general-signing structure or are already known to
 7612 the meter). Correspondingly, the values used in the OtherInput field of the KDF at section
 7613 9.2.3.4.6.5 of the Green Book are those taken from the general-signing structure, rather than
 7614 the corresponding fields in the general-ciphering structure.

7615 For ESME and GSME, the GBCS specifies particular, additional interpretation of parameters
 7616 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
<i>Vended by supplier</i>		
Originator Counter (64 bits)	10010010100011111100011100101100000000000000000000000000000000000000	10,560,878,642,999,590,912
UTRN Counter (32 bits)	10010010100011111100011100101100	2,458,896,172
PTUT Truncated UTRN Counter (10 bits)	1100101100	812
<i>Recorded on Device</i>		
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 $q = V - p$ $q = 2,458,896,167 - 807$	10010010100011111100010000000000	2,458,895,360
4	Set r = PTUT Truncated Originator Counter	1100101100	812

5	Calculate $p - 2^9$ (Call this variable, x) (See footnote 39) $x = 812 - 512$	100101100	300
6	Calculate $p + 2^9$ (Call this variable, y) $y = 812 + 512$	10100101100	1324
7	Test r against x and y and set s accordingly <ul style="list-style-type: none"> • If $r < x$ then $s = r + 2^{10}$ • If $r > y$ then $s = r - 2^{10}$ • Else $s = r$ $300 < 812 < 1324$, therefore $s = r$	1100101100	812
8	Set deduced Originator Counter = $(q + s) * 2^{32}$	100100101000111111000111001 0110000000000000000000000000 0000000000	10,560,878,642,999,590,912
9	Set deduced UTRN Counter as most significant 32 bits of Deduced Originator Counter	100100101000111111000111001 01100	2,458,896,172

Table 27: Derivation of the UTRN Counter from the PTUT Truncated UTRN Counter – a worked example

³⁹ In some cases where $p < 512$, this result may be negative. How negative binary numbers are represented in the calculation is an implementation decision, and not a matter for the GBCS since there is no impact on interoperability.

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 Devices 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

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	0x11E1A300	300,000,000

Table 28a: Data items and values to be configured prior to installation of Devices

7634 ~~This page is intentionally left blank.~~

