**MP143 ‘Incorporating IRPs into GBCS v3. Series**

This document is classified as **White** in accordance with the Panel Information Policy. Information can be shared with the public, and any members may publish the information, subject to copyright.

**Annex B**

**Legal text – version 1.0**

About this document

This document contains the redlined changes to the SEC that would be required to deliver this Modification Proposal.

These changes have been drafted against SEC Version 42.0.

This document contains the changes required to deliver the Proposed Solution.

SEC Schedule 8 – Great Britain Companion Specification (GBCS) version 3.2

These changes have been drafted against SEC Version 42.0.

These changes will be applied to version 3.2.

## Amend Section 6 Message Categories as follows:

**6.2.4 Command Authenticity and Integrity Verification**

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

***6.2.4.1 Checks to be undertaken***

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

## Amend Section 7.3.8 DLMS Device Requirements Tables as follows:

**7.3.8 DLMS Device Requirements Tables**

Table 7.3.8a: Objects tab in embedded file

Table 7.3.8b: Scripts tab in embedded file

Table 7.3.8c: Application Associations tab in embedded file

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

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

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

Table 7.3.8g: Conformance Content tab in embedded file

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



## Amend Section 10.2.2.2 GSME as follows:

***10.2.2.2 GSME***

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

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

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

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

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

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

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

For clarity, the GSME:

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

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

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

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

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

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

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

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

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

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

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

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

## Amend Section 13.2.3.1 Summary - Informative as follows:

**Common Requirements**

***Summary – informative***

Remote Party Security Credentials are provided to Devices as Certificates which are X.509 based, DER encoded ASN.1 structures. Hence, the Command’s structure is specified using ASN.1 with DER encoding to be applied to Command instances. Note that the details provided in the Response include the related Execution Counter details held on the Device.

## Amend Section 13.2.3.3 The @ProvideSecurityCredentialDetails.Command and @ProvideSecurityCredentialDetails.Response structure definition as follows:

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

Each instance of @ProvideSecurityCredentialDetails.Command and of @ProvideSecurityCredentialDetails.Response shall be an octet string containing the DER[[1]](#footnote-2) encoding of the populated structure defined in this Section 13.2.3.3 which specifies the structure in ASN.1 notation[[2]](#footnote-3).

ProvideSecurityCredentialDetails DEFINITIONS ::= BEGIN

Command ::= SEQUENCE

{

-- Identify which of the Public Keys on the Device is to be used in verifying the Signature or MAC

-- (so defining the nature of the verification by way of the KeyUsage parameter held on the

-- Device for the Public Key so identified).

authorisingRemotePartyTACellIdentifier TrustAnchorCellIdentifier,

-- List the Remote Party Role(s) for which credential details are required

remotePartyRolesCredentialsRequired SEQUENCE OF RemotePartyRole

}

Response ::= SEQUENCE OF RemotePartyDetails

RemotePartyDetails ::= SEQUENCE

{

-- Which Remote Party do these details relate to?

remotePartyRole RemotePartyRole,

-- statusCode shall be success unless the role is not valid on this type of Device or there is a processing failure

statusCode StatusCode,

-- What is the current Update Security Credentials Execution Counter on the Device for this role, where there is such a number for this role (see Table 13.2.4.4)?

currentSeqNumber SeqNumber OPTIONAL,

-- What are the details held on the Device for each of the Cells related to this role? The list shall have between one and

-- three entries (e.g. there will be one if role is transitional change of supplier; there may be three if role is supplier)

trustAnchorCellsDetails SEQUENCE OF TrustAnchorCellContents OPTIONAL

}

SeqNumber ::= INTEGER (0.. 18446744073709551615)

TrustAnchorCellContents ::= SEQUENCE

{

-- To what cryptographic use can the Public Key in this Cell be put? Some Remote Party Roles

-- (e.g. supplier) can have more than one Public Key on a Device and each one would only have

-- a single cryptographic use.

trustAnchorCellKeyUsage KeyUsage,

-- trustAnchorCellUsage is to allow for multiple Public Keys of the same keyUsage for the same Remote

-- Party Role. This will be absent except where used to refer to the Supplier Key Agreement Key.

-- This Key is used solely in relation to validating Supplier generated MACs on Prepayment Top Up transactions.

trustAnchorCellUsage CellUsage DEFAULT management,

-- The existingSubjectUniqueID shall be the 64 bit Entity Identifier of the Security Credentials in this Trust Anchor Cell.

existingSubjectUniqueID OCTET STRING,

-- The APKI requirements mean that KeyIdentifier attributes will all be 8 byte SHA-1 Hashes.

-- existingSubjectKeyIdentifier shall be set accordingly based on the contents of the Trust Anchor Cell

existingSubjectKeyIdentifier OCTET STRING

}

TrustAnchorCellIdentifier ::= SEQUENCE

{

-- Which Remote Party Role does this Cell relate to?

trustAnchorCellRemotePartyRole RemotePartyRole,

-- To what cryptographic use can the Public Key in this Cell be put? Some Remote Party Roles

-- (e.g. supplier) can have more than one Public Key on a Device and each one would only have

-- a single cryptographic use.

trustAnchorCellKeyUsage KeyUsage,

-- trustAnchorCellUsage is to allow for multiple Public Keys of the same keyUsage for the same Remote

-- Party Role. This may be absent except where use to refer to the Supplier Key

-- Agreement Key used solely in relation to validating Supplier generated MACs on Prepayment Top Up transactions

trustAnchorCellUsage CellUsage DEFAULT management

}

CellUsage ::= INTEGER {management(0), prePaymentTopUp(1)}

RemotePartyRole ::= INTEGER

{

-- Define the full set of Remote Party Roles in relation to which a Device may need to undertake

-- processing. Note that most Devices will only support processing in relation to a subset of these.

root (0),

recovery (1),

supplier (2),

networkOperator (3),

accessControlBroker (4),

transitionalCoS (5),

wanProvider (6),

issuingAuthority (7), -- Devices will receive such Certificates but they do not

-- need to store them over an extended period

-- The ‘other’ RemotePartyRole is for a party whose role does not allow it to invoke any Device function apart from

-- UpdateSecurityCredentials. This is to allow for Device functionality to be locked out of usage until a valid

-- Remote Party can be identified e.g. where roles cannot be fixed until a Device is bought in to operation

other (127)

}

-- KeyUsage is only repeated here for ease of reference. It is defined in RFC 5912

KeyUsage ::= BIT STRING

{

-- Define valid uses of Public Keys.

digitalSignature (0),

contentCommitment (1), -- not valid for GBCS compliant transactions

keyEncipherment (2), -- not valid for GBCS compliant transactions

dataEncipherment (3),

keyAgreement (4),

keyCertSign (5),

cRLSign (6),

encipherOnly (7),

decipherOnly (8) -- not valid for GBCS compliant transactions

}

-- The GBCS only allows for a constrained set of Trust Anchor Cell operations and so the list of possible outcomes

-- is more limited than in IETF RFC 5934. The list below is that more constrained subset

StatusCode ::= ENUMERATED {

success (0),

-- trustAnchorNotFound indicates that details of a trust anchor were requested, but the referenced trust anchor

-- is not represented on the Device

trustAnchorNotFound (25),

other (127)}

END

## Amend Section 13.2.4.4 Response Construction as follows:

***13.2.4.4 Response Construction***

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

The @ProvideSecurityCredentialDetails.Response shall have the structure defined in Section 13.2.3.3, and the Device shall populate with values according to Table 13.2.4.4.

| **Attribute name** | **Data Type** | **Value (blank cells mean the command specific value is derived by the encoding process)** | **Mandatory, OPTIONAL or DEFAULT value** | **Notes** |
| --- | --- | --- | --- | --- |
| @ProvideSecurityCredentialDetails.Response ::= | SEQUENCE OF |  |  |  |
| SEQUENCE |  |  |  |  |
| remotePartyRole | INTEGER | root (0) ,  recovery (1) ,  supplier (2) ,  networkOperator (3) ,  accessControlBroker (4) ,  transitionalCoS (5) ,  wanProvider (6) , | Mandatory if SEQUENCE is present | The role to which the credentials in this SEQUENCE relate |
| statusCode | ENUMERATED | success (0) ,  trustAnchorNotFound (25) ,  other (127) | Mandatory if SEQUENCE is present | Whether the Device can supply the details |
| currentSeqNumber | INTEGER | The corresponding Counter value | Present if statusCode=0 | The Execution Counter held by the Device for this role’s use of the Update Security Credentials Command. Where this role is root, the value of the anyByContingency Execution Counter shall be returned: where this role is transitionalCoS, the value of the transCoSByTransCos Execution Counter shall be returned. |
| trustAnchorCellsDetails | SEQUENCE OF |  | At least one in the SEQUENCE OF must be present if statusCode=0 |  |
| SEQUENCE |  |  |  |  |
| trustAnchorCellKeyUsage | BIT STRING | digitalSignature (0) ,  keyAgreement (4) ,  keyCertSign (5) | Mandatory if SEQUENCE is present | To what use can the public key in this Cell be put |
| trustAnchorCellUsage | INTEGER | prePaymentTopUp(1) | DEFAULT management (0) | Only needs to be present for the {supplier, keyAgreement, prePaymentTopUp} Cell |
| existingSubjectUniqueID | OCTET STRING | Entity Identifier in this Cell | Mandatory if SEQUENCE is present | See Section 12.4 |
| existingSubjectKeyIdentifier | OCTET STRING | Key Identifier of the key in this Cell | Mandatory if SEQUENCE is present |  |

Table 13.2.4.4: Attribute values for Provide Security Credentials Response

## Amend Section 13.3.4.1 Common Payload construction as follows:

***13.3.4.1 Command Payload construction***

The @UpdateSecurityCredentials.CommandPayload shall have the structure defined in Section 13.3.5.11, and the Remote Party constructing the Command shall populate with values according to Table 13.3.4.1.

| **Attribute name** | **Data Type** | **Value (blank cells mean the command specific value is derived by the encoding process)** | **Mandatory, OPTIONAL or DEFAULT value** | **Notes** |
| --- | --- | --- | --- | --- |
| @ UpdateSecurityCredentials.Command ::= | SEQUENCE |  |  |  |
| authorisingRemotePartyControl | SEQUENCE |  |  | This structure provides details to allow the Device to identify the Remote Party Role authorising this Command, check whether the rest of the payload is allowable and allow counters / counter caches on the Device to be reset, if the command changes the Remote Party in control |
| credentialsReplacementMode | INTEGER | supplierBySupplier (2) ,  networkOperatorByNetworkOperator (3),  accessControlBrokerByACB (4) ,  wanProviderByWanProvider (5) ,  transCoSByTransCoS (6) ,  supplierByTransCoS (7) ,  anyExceptAbnormalRootByRecovery (8) ,  anyByContingency (9) | Mandatory | Specify the replacement mode so that the Device can check that the Remote Party Role authorising the command is allowed to authorise this type of replacement(s) and that all replacements in the payload are allowed within this replacement mode. The structure of the label is *kindOfCertificate(s)BeingReplacedBypartydoingthereplacement .* For example, supplierBySupplier is where a new supplier Certificate is being provided to the Device by its Supplier |
| plaintextSymmetricKey | [0] IMPLICIT OCTET STRING | The symmetric key that will decrypt the encrypted Contingency Key held on the Device | OPTIONAL | Only to be present if the Contingency Key arrangements are being used (so if credentialsReplacementMode = anyByContingency). The contents provide the symmetric key to decrypt the Contingency Public Key in the (root, digitalSignature, management) Trust Anchor Cell |
| applyTimeBasedCPVChecks | [1] IMPLICIT INTEGER | disapply(1) | DEFAULT apply | Only to be present if the Remote Party sending the Command is instructing the Device not to apply time based checks as part of Certification Path Validation. This should only be in exceptional circumstances (e.g. supplier credentials on the Device have expired without replacement for unforeseen reasons) |
| authorisingRemotePartyTACellIdentifier | [2] IMPLICIT SEQUENCE |  | OPTIONAL | This structure identifies which Public Key on the Device is to be used in verifying KRP Signature. The key is identified by way of Trust Anchor Cell and so the nature of the check, by way of the KeyUsage parameter, is also identified. ‘authorisingRemotePartyTACellIdentifier’ can only be omitted when the Access Control Broker is changing its own Key Agreement credentials |
| trustAnchorCellRemotePartyRole | INTEGER | root (0),  recovery (1) ,  supplier (2) ,  networkOperator (3) ,  accessControlBroker (4) ,  transitionalCoS (5) ,  wanProvider (6) | 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 populated with the symmetric key required to decrypt that public key |
| trustAnchorCellKeyUsage | BIT STRING | digitalSignature (0) if credentialsReplacementMode <> anyByContingency,  keyCertSign (5) if credentialsReplacementMode = anyByContingency | Mandatory if authorisingRemotePartyTACellIdentifier  present | KRP Signature is a digital signature |
| trustAnchorCellUsage | INTEGER | management(0) | DEFAULT management | Must be absent since the prePaymentTopUp key pair cannot be used in relation to this Command |
| authorisingRemotePartySeqNumber | [3] IMPLICIT INTEGER | Originator Counter of Remote Party authorising the Command | Mandatory | Specify the Originator Counter for the Remote Party applying KRP Signature, or (for the Access Control Broker changing its credentials) the Access Control Broker’s Originator Counter |
| newRemotePartyFloorSeqNumber | [4] IMPLICIT INTEGER | Originator Counter of Remote Party who will have control of this Remote Party Role if 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 example is the Prepayment Top Up Command on ESME and GSME. The SpecialistSeqNumber structure allows such Counters to also be reset on change of control. Should only be present if this Command changes supplier credentials and the new supplier uses different counters for its Prepayment Top Ups than it does for other Commands |
| SEQUENCE |  |  |  |  |
| seqNumberUsage | INTEGER | prepaymentTopUp (0) | Mandatory if newRemotePartySpecialistFloorSeqNumber present | Specify the usage of the SeqNumber |
| seqNumber | INTEGER | Relevant Originator Counter | OPTIONAL | Specify the associated SeqNumber |
| otherRemotePartySeqNumberChanges | [6] IMPLICIT SEQUENCE OF |  | OPTIONAL | In some cases, one party acting in one Remote Party Role may be replacing certificates for a different Remote Party Role (e.g. transitionalCoS changing Supplier Credentials). In such cases, Execution Counters need also to be reset for that other Remote Party Role |
| SEQUENCE |  |  |  |  |
| otherRemotePartyRole | INTEGER | supplier (2) ,  networkOperator (3) ,  accessControlBroker (4) ,  transitionalCoS (5) ,  wanProvider (6) , | Mandatory if otherRemotePartySeqNumberChanges present | The Remote Party Role of the party whose credentials are being placed on the Device but which 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 |
| newRemotePartySpecialistFloorSeqNumber | SEQUENCE OF |  | OPTIONAL | Should only be present if otherRemotePartyRole = supplier, and that new supplier uses different counters to prevent replay on Prepayment Top Up |
| SEQUENCE |  |  |  |  |
| seqNumberUsage | INTEGER | prepaymentTopUp (0) | Mandatory if 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 ‘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) ,  keyCertSign (5)} , | Mandatory if SEQUENCE is present | To what use can the public key in this Cell be put |
| trustAnchorCellUsage | INTEGER | prePaymentTopUp(1)} | DEFAULT management | Should be absent unless:   * the deviceType is eSME or gSME; and * the supplier operating the Device wishes to use prepayment top up functionality on the Device, and this is a replacement of the corresponding certificate. Note the certificate specified for use in the {supplier, keyAgreement, prePaymentTopUp} Trust Anchor Cell may be the same key as that specified for the {supplier, keyAgreement, management} Trust Anchor Cell or may be different. |
| certificationPathCertificates | SEQUENCE OF Certificate | The list of certificates needed for Certification Path Validation | At least one Certificate must be present | Provide the certificates needed to undertake Certification Path Validation against the root public key held on the Device. The number of these may be less than the number of replacement certificates (e.g. a supplier may replace all of its certificates but may only need to supply one Certification Authority Certificate to link them all back to root |
| executionDateTime | GeneralizedTime | The date-time at which the replacements are to be used in updating the Device's Security Credentials | OPTIONAL | This field may only be present if credentialsReplacementMode is either supplierBySupplier  or supplierByTransCoS |

Table 13.3.4.1: Attribute values for Update Security Credentials Command

## Amend Section 13.3.4.3 Command Processing as follows:

***13.3.4.3 Command Processing***

Before undertaking any further processing, the Device shall update Execution Counter to the value of authorisingRemotePartySeqNumber.

If executionDateTime is present then the Device shall:

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

If executionDateTime is not present then the Device shall:

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

## Amend Section 13.3.4.6 executionOutcome construction as follows:

***13.3.4.6 executionOutcome construction***

| **Attribute name** | **Data Type** | **Value (blank cells mean the command specific value is derived by the encoding process)** | **Mandatory, OPTIONAL or DEFAULT value** | **Notes** |
| --- | --- | --- | --- | --- |
| executionOutcome | SEQUENCE |  |  |  |
| authorisingRemotePartySeqNumber | INTEGER | Originator Counter of Remote Party authorising the Command, as specified in the corresponding Command | Mandatory | This is to allow the Alert to be linked to the Command that caused execution |
| credentialsReplacementMode | INTEGER | supplierBySupplier (2) ,  networkOperatorByNetworkOperator (3) ,  accessControlBrokerByACB (4) ,  wanProviderByWanProvider (5) ,  transCoSByTransCoS (6) ,  supplierByTransCoS (7) ,  anyExceptAbnormalRootByRecovery (8) ,  anyByContingency (9)} , | Mandatory | Provide details of the corresponding Command that are not in the standard GBCS message header. Specifically the mode in which the Command was invoked |
| remotePartySeqNumberChanges | SEQUENCE OF |  | Mandatory containing zero, one or many occurrences of the following structure | The resulting changes to Execution Counters held on the Device |
| SEQUENCE |  |  |  |  |
| otherRemotePartyRole | INTEGER | root (0) ,  recovery (1) ,  supplier (2) ,  networkOperator (3) ,  accessControlBroker (4) ,  transitionalCoS (5) ,  wanProvider (6) , | Mandatory if SEQUENCE is present | The role which has had its Execution Counter values changed on the Device |
| otherRemotePartyFloorSeqNumber | INTEGER | The corresponding Execution Counter value | Mandatory if SEQUENCE is present |  |
| newRemotePartySpecialistFloorSeqNumber | SEQUENCE OF |  | OPTIONAL | Only present where Remote Party Role is supplier |
| SEQUENCE |  |  |  |  |
| seqNumberUsage | INTEGER | {prepaymentTopUp (0)} , | Mandatory if newRemotePartySpecialistFloorSeqNumber present | Specify the usage of the SeqNumber |
| seqNumber | INTEGER |  | Mandatory if newRemotePartySpecialistFloorSeqNumber present | Specify the associated SeqNumber |
| replacementOutcomes | SEQUENCE OF |  | One per replacement in the corresponding Command so at least one | For each replacement in the Command, detail the outcome and impacted parties |
| SEQUENCE |  |  |  |  |
| affectedTrustAnchorCell | SEQUENCE |  | Mandatory if SEQUENCE is present | Specify which Trust Anchor Cell was the target of this replacement |
| trustAnchorCellRemotePartyRole | INTEGER | root (0) ,  recovery (1) ,  supplier (2) ,  networkOperator (3) ,  accessControlBroker (4) ,  transitionalCoS (5) ,  wanProvider (6) | Mandatory if SEQUENCE is present | Specify the Remote Party Role to which the Trust Anchor Cell relates |
| trustAnchorCellKeyUsage | BIT STRING | digitalSignature (0) ,  keyAgreement (4) ,  keyCertSign (5) | Mandatory if SEQUENCE is present | To what use can the public key in this Cell be put |
| trustAnchorCellUsage | INTEGER | {management(0) ,  prePaymentTopUp(1)} | DEFAULT management | Absent unless:   * the deviceType is eSME or gSME; and * the supplier operating the Device wishes to use prepayment top up functionality on the Device, and this is a replacement of the corresponding certificate. Note the certificate specified for use in the {supplier, keyAgreement, prePaymentTopUp} Trust Anchor Cell may be the same key as that specified for the {supplier, keyAgreement, management} Trust Anchor Cell or may be different. |
| statusCode | ENUMERATED | success (0) ,  badCertificate (5) ,  noTrustAnchor (10) ,  insufficientMemory (17) ,  resourcesBusy (30) ,  other (127) | Mandatory if SEQUENCE is present | Whether the replacement to this Cell was successful or, if it failed, why it failed |
| existingSubjectUniqueID | OCTET STRING |  | Mandatory if SEQUENCE is present | The 64 bit Entity Identifier of the Remote Party whose credentials were in this Cell prior to receipt of the corresponding Command |
| existingSubjectKeyIdentifier | OCTET STRING |  | Mandatory if SEQUENCE is present | For the public key in this Cell prior to receipt of the corresponding Command |
| replacingSubjectUniqueID | OCTET STRING |  | Mandatory if SEQUENCE is present | The 64 bit Entity Identifier of the Remote Party whose credentials were to be placed in this Cell |
| replacingSubjectKeyIdentifier | OCTET STRING |  | Mandatory if SEQUENCE is present | For the public key which was to be placed in this Cell |

Table 13.3.4.6: Attribute values for executionOutcome

## Amend Section 13.3.5.1 Update Security Credentials Command Verification as follows:

***13.3.5.1 Update Security Credentials Command Verification***

The Device shall undertake the checks set out in this Section 13.3.5.1 before undertaking any other processing of the Command. The checks may be carried out in any order. Checking shall cease at the point that any one check fails. The checks required are shown in Table 13.3.5.1.

|  |  |  |
| --- | --- | --- |
| **Check Number** | **Criteria that must be tested by the Device** | **How the Device shall test the Criteria** |
| 1.1 | The Message is for the Device | The value of the Business Target ID in the Grouping Header in Command instance must be equal to the Device’s Entity Identifier |
| 1.2 | The Message Code is for Update Security Credentials | The value in the Message Code field of the Grouping Header must be equal to the value specified in Table 13.3.5.2 for the CredentialsReplacementMode specified in CommandPayload |
| 1.3 | If executionDateTime is present the Command is to replace Supplier Security Credentials. | If executionDateTime is present then credentialsReplacementMode must either supplierBySupplier  or supplierByTransCoS |
| 1.4 | The Device has not already actioned this Command. | As specified in Section 13.3.5.3 |
| 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 |

Table 13.3.5.1: Update Security Credentials Command authenticity and integrity verification

## Amend Section 13.3.5.3 Preventing Replay of Commands as follows:

***13.3.5.3 Preventing Replay of Commands***

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

For each of RemotePartyRole from which the Device can receive a valid Updated Security Credentials Command, the Device shall allocate storage for an Execution Counter which shall be capable of storing a 64 bit unsigned integer and which shall initially be set to the value zero at manufacture. For transitionalCoS, the Device may allocate storage for an additional Execution Counter, credentialsReplacementModes.

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

* (authorisingRemotePartyTACellIdentifier is populated in the Command) and (the authorisingRemotePartySeqNumber is strictly numerically greater than the Execution Counter the Device has recorded for the RemotePartyRole identified in authorisingRemotePartyTACellIdentifier and, where relevant for transitionalCoS, the credentialsReplacementMode);

else

* (the authorisingRemotePartySeqNumber is strictly numerically greater than the Execution Counter the Device has recorded for the accessControlBroker).

## Amend Section 13.3.5.8 Verifying the Cryptographic Protections as follows:

***13.3.5.8 Verifying the Cryptographic Protections***

In verifying Cryptographic Protections pursuant to this Section 13.3.5.8:

* KRP Signature shall, where required by Section 13.3.3.1 for the specified Message Code and credentialsReplacementMode, be verified according to the requirements in this Section 13.3.5.8; and
* ACB-SMD MAC, where required by Section 13.3.3.1 for the specified Message Code and credentialsReplacementMode, shall be verified according to the requirements in Section 6.2.4.1.2.

If credentialsReplacementMode = anyByContingency or Message Code = 0x0109 then KRP Signature shall be verified using the public key established according to the requirements of Section 13.3.5.8.1.

If credentialsReplacementMode = <> anyByContingency or Message Code = <> 0x0109 then KRP Signature shall be verified using the public key identified as per Section 4.3.2.7.2.

If credentialsReplacementMode = accessControlBrokerByACB or Message Code = 0x0109 and deviceType is not communicationsHubCommunicationsHubFunction then ACB-SMD MAC shall be verified as per Section 6.2.4.1.2.

## Amend Section 13.3.5.11 The@UpdateSecurityCredentials.CommandPayload,@UpdateSecurityCredentials.ResponsePayload and @UpdateSecurityCredentials.AlertPayload structure definition as follows:

***13.3.5.11 The @UpdateSecurityCredentials.CommandPayload,@UpdateSecurityCredentials.ResponsePayload and @UpdateSecurityCredentials.AlertPayload structure definition***

Each instance of @UpdateSecurityCredentials.CommandPayload, @UpdateSecurityCredentials.ResponsePayload and of @UpdateSecurityCredentials.AlertPayload shall be an octet string containing the DER encoding of the populated structure defined in this Section 13.3.4, which specifies the structure in ASN.1.

The structure of Certificate shall be as defined in ASN.1 in IETF RFC 5912. Note that the Certificate structures within IETF RFC 5912 begin after the phrase ‘Certificate- and CRL-specific structures begin here’.

UpdateSecurityCredentials DEFINITIONS ::= BEGIN

CommandPayload ::= SEQUENCE

{

-- Provide details to allow the Device to identify the Remote Party Role authorising

-- this Command, check whether the rest of the payload is allowable, prevent replay attacks

-- and allow counters / counter caches on the Device to be reset, if the Command changes the Remote Party

-- in control.

-- The Remote Party authorising the Command is that party which generated the KRP Signature (or the Access Control Broker

-- if there is no KRP Signature)

authorisingRemotePartyControl AuthorisingRemotePartyControl,

-- One TrustAnchorReplacement structure is required for each Trust Anchor Cell that is to be updated

replacements SEQUENCE OF TrustAnchorReplacement,

-- Provide the certificates needed to undertake Certification Path Validation of the new

-- end entity certificate against the root public key held on the Device. The number of these may be less

-- than the number of replacement certificates (e.g. a supplier may replace all of its certificates but

-- may only need to supply one Certification Authority Certificate to link them all back to the root public

-- key as currently stored on the Device.

certificationPathCertificates SEQUENCE OF Certificate,

-- If the Command is to be future dated, specify the date-time at which the certificate replacement is to happen

executionDateTime GeneralizedTime OPTIONAL

}

ResponsePayload ::= SEQUENCE

{

-- if the Command is future dated, the Response will not have any details of execution (those will be in the subsequent alert)

commandAccepted NULL,

-- if the Command is for immediate execution, the Response will detail the outcomes

executionOutcome ExecutionOutcome OPTIONAL

}

AlertPayload ::= SEQUENCE

{

-- specify the Alert Code

alertCode INTEGER(0..4294967295),

-- specify the date-time of execution

executionDateTime GeneralizedTime,

-- detail what happened when the future dated Command was executed

executionOutcome ExecutionOutcome

}

ExecutionOutcome ::= SEQUENCE

{

-- Provide details of the corresponding Command that may not be in the standard GBCS message header. Specifically the

-- mode in which the Command was invoked, the Originator Counter in the original Command and the resulting changes to any

-- replay counters held on the Device

authorisingRemotePartySeqNumber SeqNumber,

credentialsReplacementMode CredentialsReplacementMode,

remotePartySeqNumberChanges SEQUENCE OF RemotePartySeqNumberChange,

-- For each replacement in the Command, detail the outcome and impacted parties

replacementOutcomes SEQUENCE OF ReplacementOutcome

}

AuthorisingRemotePartyControl ::= SEQUENCE

{

-- Specify the replacement mode so that the Device can check that the Remote Party Role is allowed to

-- authorise this type of replacement and that all replacements in the payload are allowed within this

-- replacement mode

credentialsReplacementMode CredentialsReplacementMode,

-- Only if credentialsReplacementMode = anyByContingency, provide the symmetric key to decrypt

-- the Contingency Public Key in the (root, digitalSignature, management) Trust Anchor Cell

plaintextSymmetricKey [0] IMPLICIT OCTET STRING OPTIONAL,

-- Specify whether the time based checks as part of any Certificate Path Validation should be applied

applyTimeBasedCPVChecks [1] IMPLICIT INTEGER {apply(0), disapply(1)} DEFAULT apply,

-- Identify which of the Public Keys on the Device is to be used in checking KRP Signature

-- ‘authorisingRemotePartyTACellIdentifier’ may only be omitted when

-- the access control broker is updating its own credentials and the target device is not a CHF.

-- In all other cases it is mandatory.

authorisingRemotePartyTACellIdentifier [2] IMPLICIT TrustAnchorCellIdentifier OPTIONAL,

-- Specify the Originator Counter for the Remote Party Applying KRP Signature, or (for the

-- Access Control Broker changing its credentials) the Access Control Broker’s Originator Counter.

authorisingRemotePartySeqNumber [3] IMPLICIT SeqNumber,

-- If the Command is to effect a change of control, then newRemotePartyFloorSeqNumber must be included

-- and will be the value used to prevent replay of Update Security Credentials Commands for the

-- new controlling Remote Party.

newRemotePartyFloorSeqNumber [4] IMPLICIT SeqNumber OPTIONAL,

-- Some Commands on the Device may use a different Originator Counter sequence for Protection Against Replay. At this

-- version of the GBCS, the only example is the Prepayment Top Up Command on ESME and GSME. The

-- SpecialistSeqNumber structure allows such Counters to also be reset on change of control.

newRemotePartySpecialistFloorSeqNumber [5] IMPLICIT SEQUENCE OF SpecialistSeqNumber OPTIONAL,

-- In some cases, one party acting in one Remote Party Role may be replacing certificates for a different Remote Party Role.

-- In some cases, Execution Counters need also to be reset for those other Remote Party Role(s)

otherRemotePartySeqNumberChanges [6] IMPLICIT SEQUENCE OF RemotePartySeqNumberChange OPTIONAL

}

RemotePartySeqNumberChange ::= SEQUENCE

{

otherRemotePartyRole RemotePartyRole,

otherRemotePartyFloorSeqNumber SeqNumber,

newRemotePartySpecialistFloorSeqNumber SEQUENCE OF SpecialistSeqNumber OPTIONAL

}

SpecialistSeqNumber ::= SEQUENCE

{

-- Specify the usage of the SeqNumber

seqNumberUsage SeqNumberUsage,

-- Specify the associated SeqNumber

seqNumber SeqNumber

}

SeqNumberUsage ::= INTEGER

{

-- Define the full set of discrete usages on a Device. The only specialist

-- counter is for Prepayment Top Up (which is set independently of other counters). This may only be

-- included when changing Supplier Security Credentials on an ESME or GSME.

prepaymentTopUp (0)

}

SeqNumber ::= INTEGER (0.. 18446744073709551615)

TrustAnchorReplacement ::= SEQUENCE

{

-- Provide the new end entity certificate

replacementCertificate Certificate,

-- Specify where it is to go (specifically which Trust Anchor Cell is to have its details replaced using

-- the new end entity certificate)

targetTrustAnchorCell TrustAnchorCellIdentifier

}

ReplacementOutcome ::= SEQUENCE

{

affectedTrustAnchorCell TrustAnchorCellIdentifier,

statusCode StatusCode,

-- The GBCS Certificate requirements mean that the Subject Unique ID attribute in the subject field of a certificate will always

-- contain the 64 bit unique number that equates to Entity Identifier. existingSubjectUniqueID should be set

-- accordingly based on the contents of the Trust Anchor Cell prior to Command processing.

existingSubjectUniqueID OCTET STRING,

-- The GBCS Certificate requirements mean that subjectKeyIdentifier attributes will all be 8 byte SHA-1 Hashes.

-- existingSubjectKeyIdentifier should be set accordingly based on the contents of the Trust Anchor Cell prior to

-- Command processing.

existingSubjectKeyIdentifier OCTET STRING,

-- The Subject Unique ID in the subject field of the certificate in this TrustAnchorReplacement

replacingSubjectUniqueID OCTET STRING,

-- The subjectKeyIdentifier in the certificate in this TrustAnchorReplacement

replacingSubjectKeyIdentifier OCTET STRING

}

TrustAnchorCellIdentifier ::= SEQUENCE

{

-- Which Remote Party Role does this Cell relate to?

trustAnchorCellRemotePartyRole RemotePartyRole,

-- To what cryptographic use can the Public Key in this Cell be put? Some Remote Party Roles

-- (e.g. supplier) can have more than one Public Key on a Device and each one would only have

-- a single cryptographic use.

trustAnchorCellKeyUsage KeyUsage,

-- trustAnchorCellUsage is to allow for multiple Public Keys of the same keyUsage for the same Remote

-- Party Role. It will be absent except where used to refer to the Supplier Key

-- Agreement Key used solely in relation to validating Supplier generated MACs on Prepayment Top Up

-- transactions

trustAnchorCellUsage CellUsage DEFAULT management

}

CellUsage ::= INTEGER {management(0), prePaymentTopUp(1)}

RemotePartyRole ::= INTEGER

{

-- Define the full set of Remote Party Roles in relation to which a Device may need to undertake

-- processing. Note that most Devices will only support a subset of these.

root (0),

recovery (1),

supplier (2),

networkOperator (3),

accessControlBroker (4),

transitionalCoS (5),

wanProvider (6),

issuingAuthority (7), -- Devices will receive such Certificates but they do not need to store

-- them over an extended period

-- The ‘other’ RemotePartyRole is for a party whose role does not allow it to invoke any Device function apart from

-- UpdateSecurityCredentials. This is to allow for Device functionality to be locked out of usage until a valid

-- Remote Party can be identified e.g. where roles cannot be fixed until a Device is brought in to operation

other (127)

}

-- KeyUsage is only repeated here for clarity. It is defined in RFC 5912

KeyUsage ::= BIT STRING

{

-- Define valid uses of Public Keys held by Devices in their Trust Anchor Cells.

digitalSignature (0),

contentCommitment (1), -- not valid for GBCS compliant transactions

keyEncipherment (2), -- not valid for GBCS compliant transactions

dataEncipherment (3), -- not valid for GBCS compliant transactions

keyAgreement (4),

keyCertSign (5),

cRLSign (6),

encipherOnly (7), -- not valid for GBCS compliant transactions

decipherOnly (8) -- not valid for GBCS compliant transactions

}

CredentialsReplacementMode ::= INTEGER

{

-- Define the valid combinations as to which Remote Party Roles can replace which kinds of Trust Anchors.

-- Normal operational replacement modes

supplierBySupplier (2),

networkOperatorByNetworkOperator (3),

accessControlBrokerByACB (4),

wanProviderByWanProvider (5),

transCoSByTransCoS (6),

supplierByTransCoS (7),

-- Recovery modes

anyExceptAbnormalRootByRecovery (8),

anyByContingency (9)

}

-- The GBCS only allows for a constrained set of Trust Anchor Cell operations and so the list of possible outcomes

-- is more limited than in RFC 5934. The list below is that more constrained subset

StatusCode ::= ENUMERATED {

success (0),

-- badCertificate is used to indicate that the syntax for one or more certificates is invalid.

badCertificate (5),

-- noTrustAnchor is used to indicate that the authorityKeyIdentifier does not identify the public key of a

-- trust anchor or a certification path that terminates with an installed trust anchor

noTrustAnchor (10),

-- insufficientMemory indicates that the update could not be processed because the Device did not

-- have sufficient memory

insufficientMemory (17),

-- resourcesBusy indicates that the resources necessary to process the replacement are not available at the

-- present time, but the resources might be available at some point in the future.

resourcesBusy (30),

-- other indicates that the update could not be processed, but the reason is not covered by any of the assigned

-- status codes. Use of this status code SHOULD be avoided.

other (127) }

END

## Amend Section 13.3.5.12 Requirements for AuthorisingRemotePartyControl elements – informative as follows:

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

All bar two parts of the AuthorisingRemotePartyControl structure are optional. This Section summarises when each of the optional elements needs to be present.

| AuthorisingRemotePartyControl **element** | **Notes** |
| --- | --- |
| credentialsReplacementMode | Always required |
| plaintextSymmetricKey | Only required if credentialsReplacementMode = anyByContingency (when it is always required) |
| applyTimeBasedCPVChecks | Only required if the Device is to ignore time when undertaking Certification Path Validation, in which case it needs to have the value ‘disapply’ |
| authorisingRemotePartyTACellIdentifier | For a Communications Hub, always present.  For all other Devices, always present unless the Access Control Broker is replacing its own Key Agreement credentials (in which case it should be omitted) |
| authorisingRemotePartySeqNumber | Always required |
| newRemotePartyFloorSeqNumber | If the Command is to effect a change of control, then newRemotePartyFloorSeqNumber should be included. It can be present in all other situations |
| newRemotePartySpecialistFloorSeqNumber | Only required on Change of Supplier where the new Supplier has decided to use a different sequence of Originator Counters for prepayment top ups. |
| otherRemotePartySeqNumberChanges | Should be present if one role (e.g. recovery, transitionalCoS) is changing credentials for another role or roles (e.g. supplier). In such cases, this should be present to set Execution Counters for that other role or roles |

Table 13.3.5.12: Requirements for AuthorisingRemotePartyControl element

## Amend Section 13.5.4 Device processing of Commands and Response handling as follows:

### 13.5.4 Device processing of Commands and Response handling

The Device receiving an Update Device Certificate on Device Command shall undertake processing steps in the sequence defined in this Section 13.5.4.

In processing an Update Device Certificate on Device Command, the Device shall:

1. undertake Command Authenticity and Integrity Verification as required for a Command of Message Category SME.C.C, except that check 4 in Section 6.2.4.1.1 may be undertaken after the checks in Section 6.2.4.1.2. The Security Credentials used to verify Cryptographic Protection I shall be:

* those held in the {wANProvider, digitalSignature, management} Trust Anchor Cell, if the target Device’s deviceType equals communicationsHubCommunicationsHubFunction; or
* those held in the {supplier, digitalSignature, management} Trust Anchor Cell, if the target Device’s deviceType does not equal communicationsHubCommunicationsHubFunction.

1. establish the values of keyUsage, subjectPublicKey and hwSerialNum in certificate in the CommandPayload. If any of the values cannot be established then the Device shall set updateDeviceCertResponseCode to invalidCertificate, and process from step 10;
2. validate that hwSerialNum established at step 2 is the Device’s Entity Identifier. If this validation fails then the Device shall set updateDeviceCertResponseCode to wrongDeviceIdentity, and process from step 10;
3. validate that keyUsage established at step 2 is either digitalSignature only or keyAgreement only. If this validation fails then the Device shall set updateDeviceCertResponseCode to invalidKeyUsage, and process from step 10;
4. validate that the Device holds a Pending Private Key for the keyUsage as established at step 2. If this validation fails then the Device shall set updateDeviceCertResponseCode to noCorrespondingKeyPairGenerated, and process from step 10;
5. validate that subjectPublicKey established at step 2 is the bit string representation of the Public Key corresponding to the Pending Private Key identified at step 5. If this validation fails then the Device shall set updateDeviceCertResponseCode to wrongPublicKey, and process from step 10;
6. store certificate. If this step fails then the Device shall set updateDeviceCertResponseCode to certificateStorageFailed, and process from step 10;
7. set the Current Private Key to have the value of the Pending Private Key for the keyUsage established at step 2. If this step fails then the Device shall set updateDeviceCertResponseCode to privateKeyChangeFailed, and process from step 10;
8. set updateDeviceCertResponseCode to success; and
9. create a Response according to the requirements of Section 13.5.7, apply the Response Cryptographic Protection required for a Response of Message Category SME.C.C, and send the Response.

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

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

## Amend Section 13.7.4.2.2 Device processing of Commands and Response handling as follows:

*13.7.4.2.2 Device processing of Commands and Response handling*

The Device receiving a ‘Join Device’ Command shall undertake processing steps in the sequence defined in this Section 13.7.4.2.2. Should a step after step 1 be unsuccessful, the Device shall create a Response according to the requirements of Section 13.4.7, apply the Response Cryptographic Protection required for a Response of the relevant Message Category, and send the Response and shall not undertake any further steps defined in this Section 13.7.4.2.2.

In processing a ‘Join Device’ Command, the Device shall:

1. undertake Command Authenticity and Integrity Verification as required for a Command of this Message Category, except that check 4 in Section 6.2.4.1.1 may be undertaken after the checks in Section 6.2.4.1.2. The Security Credentials used to verify Cryptographic Protection 1 shall be:

* those held in the {accessControlBroker, digitalSignature, management} Trust Anchor Cell, if deviceType equals type1PrepaymentInterfaceDevice; or
* those held in the {supplier, digitalSignature, management} Trust Anchor Cell, if deviceType does not equal type1PrepaymentInterfaceDevice;

1. verify the joinMethodAndRole as specified in Section 13.7.4.5.3;
2. add the otherDeviceEntityIdentifier and otherDeviceType to its Device Log as specified in Section 13.7.4.5.4;
3. if deviceType is eSME then undertake Key Establishment with the other Device as specified in Section 13.7.4.5.5;
4. if joinMethodAndRole is methodC, and so the join is between a gSME and a type1PrepaymentInterfaceDevice, check that otherDeviceCertificate is present and validly structured. If the check succeeds the Device shall store, linked to this Device Log entry, details relating to otherDeviceCertificate, such that the Device is able to use subsequently the Shared Secret derived from otherDeviceCertificate and its own Private Key Agreement Key. If this check fails the Device shall set joinResponseCode to invalidOrMissingCertificate and processing shall be unsuccessful;
5. set joinResponseCode to success, create a Response according to the requirements of Section 13.4.7, apply the Response Cryptographic Protection required for a Response of the relevant Message Category, and send the Response.

## Amend Section 18.1.1.3 ZSE Get Scheduled Events command as follows:

* + - 1. ***ZSE Get Scheduled Events command***

When sending a ZSE Get Scheduled Event command pursuant to section 8.5.2.1 of SMETS, an HCALCS shall populate that ZCL Command according to Table 18.1.1.3.

On authenticated receipt of ZSE Get Scheduled Event command, the ESME shall send a ZSE Load Control Event command instructing the HCALCS whether it is to be open or closed, and for how long it is to be in that state, or send a ZSE Default Response command with Status = NOT\_FOUND if there are no settings for this HCALCS in the ESME's Auxiliary 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 |

Table 18.1.1.3: ZSE Get Scheduled Event command

## Amend Section 18.2 DLMS COSEM Message Templates as follows:

**18.2 DLMS COSEM Message Templates**

Table 18.2 contains Message Templates for all Use Case with DLMS COSEM payloads. These Message Templates are derived from the Mapping Table, and shall be complied with in the construction and population of all such Messages.



Table 18.2: DLMS COSEM Message Templates

## Amend Section 18.3 Illustrative command and response installation and DER encoding as follows:

**18.3 Illustrative command and response instantiation and DER encoding**

**18.3.1 Illustrative @UpdateSecurityCredentials.CommandPayload instantiation and its DER encoding – informative**

supplierUpdatingAllSupplierCertificates in Table 18.3.1a is an ASN.1 structured value assignment. This specific example is where a Device’s Supplier is instructing the Device to replace both the Supplier Digital Signing and Key Agreement credentials on the Device, and resetting Execution Counters. In business terms, an example of this would be at Change of Supplier.

The black text specifies the parts of the ASN.1 structure, the blue text specifies the value it is set to and the comments explain each of the values.

| **ASN.1** | **Notes** |
| --- | --- |
| supplierUpdatingAllSupplierCertificates CommandPayload ::=  {authorisingRemotePartyControl  {credentialsReplacementMode *supplierBySupplier,*  authorisingRemotePartyTACellIdentifier  {trustAnchorCellRemotePartyRole *supplier*,  trustAnchorCellKeyUsage { *digitalSignature*}},  authorisingRemotePartySeqNumber *123456789*,  newRemotePartyFloorSeqNumber *987654321*}  replacements  {{replacementCertificate *'0A7C8E9F123456789ABCDEF01234'H*,  targetTrustAnchorCell  {trustAnchorCellRemotePartyRole *supplier*,  trustAnchorCellKeyUsage { *digitalSignature*}}}  {replacementCertificate *'0B34269F123456789ABCDEF01234'*H,  targetTrustAnchorCell  {trustAnchorCellRemotePartyRole *supplier*,  trustAnchorCellKeyUsage {*keyAgreement*}}}}  certificationPathCertificates {*'FFAABB9F123456789ABCDEF01234'*H }} | This message is for the supplier replacing supplier credentials  The public key to be used to check the signature on this message is the supplier digital signing key currently held by the Device.  This is the existing supplier’s counter, so greater than any this supplier has used  This is the new supplier’s counter, which the Device should use if the Command is successful  The new supplier’s digital signing certificate …  … which is to be placed in the Device’s supplier, digital signature Trust Anchor Cell  The new supplier’s key agreement certificate…  which is to be placed in the Device’s supplier, key agreement Trust Anchor Cell  The Certificate for the CA which issued the new supplier’s certificates. The Device will use this to check that the new supplier certificates were properly issued. |

Table 18.3.1a: Illustrative @UpdateSecurityCredentials.CommandPayload instantiation – ASN.1 structure

The message sent to the Device would contain the DER encoding of the above ASN.1 value assignment. This DER encoding is laid out and explained in Table 18.3.1b. For these purposes, the Certificate is simply shown as an OCTET STRING.

| **Component** | **Value** | **Notes** |
| --- | --- | --- |
| CommandPayload SEQUENCE: |  |  |
| tag = [UNIVERSAL 16] constructed; | 0x30 | Tag for SEQUENCE |
| length = | 0x64 | 100 octet length follows |
| contents =: |  |  |
| authorisingRemotePartyControl AuthorisingRemotePartyControl SEQUENCE: |  |  |
| tag = [UNIVERSAL 16] constructed; | 0x30 | Tag for SEQUENCE |
| length = | 0x18 | Length of authorisingRemotePartyControl |
| contents =: |  |  |
| credentialsReplacementMode CredentialsReplacementMode INTEGER: |  |  |
| tag = [UNIVERSAL 2] primitive; | 0x02 |  |
| length = | 0x01 |  |
| contents =: | 0x02 | Representing supplierBySupplier |
| authorisingRemotePartyTACellIdentifier TrustAnchorCellIdentifier SEQUENCE: |  |  |
| tag = [2] constructed; | 0xA2 | Tag for authorisingRemotePartyTACellIdentifier |
| length = | 0x07 | Length of authorisingRemotePartyTACellIdentifier |
| contents =: |  |  |
| trustAnchorCellRemotePartyRole RemotePartyRole INTEGER: |  |  |
| tag = [UNIVERSAL 2] primitive; | 0x02 | Tag for INTEGER |
| length = | 0x01 | 1 octet length INTEGER |
| contents =: | 0x02 | Representing supplier RemotePartyRole |
| trustAnchorCellKeyUsage KeyUsage BIT STRING: |  |  |
| tag = [UNIVERSAL 3] primitive; | 0x03 | Tag for BIT STRING |
| length = | 0x02 | 2 octet length BIT STRING |
| contents =: | 0x0780 | Representing digitalSignature |
| authorisingRemotePartySeqNumber SeqNumber INTEGER: |  |  |
| tag = [3] primitive; | 0x83 | Tag for INTEGER |
| length = | 0x04 | 4 octet length INTEGER |
| contents =: | 0x075bcd15 | The old supplier’s Execution Counter in hex |
| newRemotePartyFloorSeqNumber SeqNumber INTEGER: |  |  |
| tag = [4] primitive; | 0x84 | Tag for INTEGER |
| length = | 0x04 | 4 octet length INTEGER |
| contents =: | 0x3ade68b1 | The new supplier’s Execution Counter in hex |
| replacements SEQUENCE OF: |  |  |
| tag = [UNIVERSAL 16] constructed; | 0x30 | Tag for SEQUENCE |
| length = | 0x36 | Length of replacements |
| contents =: |  |  |
| TrustAnchorReplacement SEQUENCE: |  |  |
| tag = [UNIVERSAL 16] constructed; | 0x30 | Tag for SEQUENCE |
| length = | 0x19 | Length of first TrustAnchorReplacement |
| contents =: |  |  |
| replacementCertificate Certificate OCTET STRING: |  |  |
| tag = [UNIVERSAL 4] primitive; | 0x04 | Tag for OCTET STRING |
| length = | 0x0e | Length of certificate |
| contents =: | 0x0a7c8e9f123456789abcdef01234 | New supplier’s digitalSignature certificate |
| targetTrustAnchorCell TrustAnchorCellIdentifier SEQUENCE: |  |  |
| tag = [UNIVERSAL 16] constructed; | 0x30 | Tag for SEQUENCE |
| length = | 0x07 | Length of targetTrustAnchorCell |
| contents =: |  |  |
| trustAnchorCellRemotePartyRole RemotePartyRole INTEGER: |  |  |
| tag = [UNIVERSAL 2] primitive; | 0x02 | Tag for INTEGER |
| length = | 0x01 | 1 octet length INTEGER |
| contents =: | 0x02 | Representing supplier RemotePartyRole |
| trustAnchorCellKeyUsage KeyUsage BIT STRING: |  |  |
| tag = [UNIVERSAL 3] primitive; | 0x03 | Tag for BIT STRING |
| length = | 0x02 | 2 octet length BIT STRING |
| contents =: | 0x0780 | Representing digitalSignature |
| TrustAnchorReplacement SEQUENCE: |  |  |
| tag = [UNIVERSAL 16] constructed; | 0x30 | Tag for SEQUENCE |
| length = | 0x19 | Length of second TrustAnchorReplacement |
| contents =: |  |  |
| replacementCertificate Certificate OCTET STRING: |  |  |
| tag = [UNIVERSAL 4] primitive; | 0x04 | Tag for OCTET STRING |
| length = | 0x0e | Length of certificate |
| contents =: | 0x0b34269f123456789abcdef01234 | New supplier’s keyAgreement certificate |
| targetTrustAnchorCell TrustAnchorCellIdentifier SEQUENCE: |  |  |
| tag = [UNIVERSAL 16] constructed; | 0x30 | Tag for SEQUENCE |
| length = | 0x07 | Length of targetTrustAnchorCell |
| contents =: |  |  |
| trustAnchorCellRemotePartyRole RemotePartyRole INTEGER: |  |  |
| tag = [UNIVERSAL 2] primitive; | 0x02 | Tag for INTEGER |
| length = | 0x01 | 1 octet length INTEGER |
| contents =: | 0x02 | Representing supplier RemotePartyRole |
| trustAnchorCellKeyUsage KeyUsage BIT STRING: |  |  |
| tag = [UNIVERSAL 3] primitive; | 0x03 | Tag for BIT STRING |
| length = | 0x02 | 2 octet length BIT STRING |
| contents =: | 0x0308 | Representing keyAgreement |
| certificationPathCertificates SEQUENCE OF: |  |  |
| tag = [UNIVERSAL 16] constructed; | 0x30 | Tag for SEQUENCE |
| length = | 0x10 | Length of certificationPathCertificates |
| contents =: |  |  |
| Certificate OCTET STRING: |  |  |
| tag = [UNIVERSAL 4] primitive; | 0x04 | Tag for OCTET STRING |
| length = | 0x0e | Length of certificate |
| contents =: | 0xffaabb9f123456789abcdef01234 | CA certificate for new supplier |

Table 18.3.1b: Illustrative @UpdateSecurityCredentials.Command instantiation – DER encoding

**18.3.2 Illustrative @UpdateSecurityCredentials.ResponsePayload instantiation and its DER encoding – informative**

supplierUpdatingAllSupplierCertificatesResponse in Table 18.3.2a is an ASN.1 structured value assignment. This specific example is where a Device is responding successfully to a Command.

The black text specifies the parts of the ASN.1 structure, the *blue text* specifies the value it is set to by the Device and the comments explain each of the values.

| **ASN.1** | Notes |
| --- | --- |
| supplierUpdatingAllSupplierCertificatesResponse ResponsePayload ::=  { commandAccepted NULL,  executionOutcome  {authorisingRemotePartySeqNumber 123456789,  credentialsReplacementMode *supplierBySupplier*,  remotePartySeqNumberChanges  {{otherRemotePartyRole *supplier*,  otherRemotePartyFloorSeqNumber *987654321*}  },  replacementOutcomes  {{  {affectedTrustAnchorCell  { trustAnchorCellRemotePartyRole *supplier*,  trustAnchorCellKeyUsage { *digitalSignature*}},  statusCode *success*,  existingSubjectUniqueID '*123456789ABCDEF0*'H,  existingSubjectKeyIdentifier '*1234567890123456*'H,  replacingSubjectUniqueID '*FEDCBA9876543210*'H,  replacingSubjectKeyIdentifier '*ABCDEABCDEABCDEA*'H},  {affectedTrustAnchorCell  {trustAnchorCellRemotePartyRole *supplier*,  trustAnchorCellKeyUsage { *keyAgreement*}},  statusCode *success*,  existingSubjectUniqueID *'123456789ABCDEF0*'H,  existingSubjectKeyIdentifier *'0987654321098765*'H,  replacingSubjectUniqueID *'FEDCBA9876543210*'H,  replacingSubjectKeyIdentifier *'FEDCBFEDCBFEDCBF*'H}}}} | The corresponding Command was for the Supplier replacing supplier credentials  This is the new supplier’s counter, which the Device will now use for Protection Against Replay in relation to the supplier role  This outcome is for the supplier digital signing store  The old supplier’s Entity Identifier  The KeyIdentifier for the old supplier’s digital signing key  The new supplier’s Entity Identifier  The KeyIdentifier for the old supplier’s digital signing key  This outcome is for the supplier key agreement store |

Table 18.3.2a: Illustrative @UpdateSecurityCredentials.Response instantiation – ASN.1 structure

The message sent by the Device would contain the DER encoding of the above ASN.1 value assignment. This DER encoding is laid out and explained in Table 18.3.2b.

| **Component** | **Value** | **Notes** |
| --- | --- | --- |
| ResponsePayload SEQUENCE: |  |  |
| tag = [UNIVERSAL 16] constructed; | 0x30 | Tag for SEQUENCE |
| length = | 0x8189 | Length 137 |
| content = |  |  |
| commandAccepted NULL |  |  |
| tag = [UNIVERSAL 5] primitive |  | 0X05 |
| length = | 0X00 |  |
| executionOutcome ExecutionOutcome SEQUENCE |  |  |
| tag = [UNIVERSAL 16] constructed | 0X30 | Tag for SEQUENCE |
| length = | 0X8184 | Length 132 |
| content = |  |  |
| authorisingRemotePartySeqNumber SeqNumber INTEGER: |  |  |
| tag = [UNIVERSAL 2] primitive | 0x02 | Tag for INTEGER |
| length = | 0x04 | 4 octet length INTEGER |
| contents = | 0X075BCD15 | The old supplier’s Execution Counter in hex |
| credentialsReplacementMode CredentialsReplacementMode INTEGER: |  |  |
| tag = [UNIVERSAL 2] primitive; | 0x02 | Tag for INTEGER |
| length = | 0x01 |  |
| content = | 0x02 | Value for supplierBySupplier |
| remotePartySeqNumberChanges SEQUENCE OF: |  |  |
| tag = [UNIVERSAL 16] constructed; | 0x30 | Tag for SEQUENCE |
| length = | 0x0B |  |
| content = |  |  |
| RemotePartySeqNumberChange SEQUENCE: |  |  |
| tag = [UNIVERSAL 16] constructed; | 0x30 | Tag for SEQUENCE |
| length = | 0x09 |  |
| content = |  |  |
| otherRemotePartyRole RemotePartyRole INTEGER: |  |  |
| tag = [UNIVERSAL 2] primitive; | 0x02 | Tag for INTEGER |
| length = | 0x01 |  |
| content = | 0x02 | Value for supplier |
| otherRemotePartyFloorSeqNumber SeqNumber INTEGER: |  |  |
| tag = [UNIVERSAL 2] primitive; | 0x02 | Tag for INTEGER |
| length = | 0x04 |  |
| content = | 0x3ade68b1 | The new supplier’s Execution Counter in hexadecimal |
| replacementOutcomes SEQUENCE OF: |  |  |
| tag = [UNIVERSAL 16] constructed; | 0x30 | Tag for SEQUENCE |
| length = | 0x6C | Length of 108 |
| content = |  |  |
| ReplacementOutcome SEQUENCE: |  |  |
| tag = [UNIVERSAL 16] constructed; | 0x30 | Tag for SEQUENCE |
| length = | 0x34 | Length of 52 |
| content = |  |  |
| affectedTrustAnchorCell TrustAnchorCellIdentifier SEQUENCE: |  |  |
| tag = [UNIVERSAL 16] constructed; | 0x30 | Tag for SEQUENCE |
| length = | 0x07 |  |
| content = |  |  |
| trustAnchorCellRemotePartyRole RemotePartyRole INTEGER: |  |  |
| tag = [UNIVERSAL 2] primitive; | 0x02 | Tag for INTEGER |
| length = | 0x01 |  |
| content = | 0x02 | Value for supplier |
| trustAnchorCellKeyUsage KeyUsage BIT STRING: |  |  |
| tag = [UNIVERSAL 3] primitive; | 0x03 | Tag for BIT STRING |
| length = | 0x02 |  |
| content = | 0x0780 | Tag for digitalSignature |
| statusCode StatusCode ENUMERATED: |  |  |
| tag = [UNIVERSAL 10] primitive; | 0x0A | Tag for ENUMERATED |
| length = | 0x01 |  |
| content = | 0x00 | Value for success |
| existingSubjectUniqueID OCTET STRING: |  |  |
| tag = [UNIVERSAL 4] primitive; | 0x04 | Tag for OCTET STRING |
| length = | 0x08 | 8 octet length of Entity Identifier |
| content = | 0x123456789abcdef0 |  |
| existingSubjectKeyIdentifier OCTET STRING: |  |  |
| tag = [UNIVERSAL 4] primitive; | 0x04 | Tag for OCTET STRING |
| length = | 0x08 | length of KeyIdentifier |
| content = | 0x1234567890123456 | KeyIdentifier |
| replacingSubjectUniqueID OCTET STRING: |  |  |
| tag = [UNIVERSAL 4] primitive; | 0x04 | Tag for OCTET STRING |
| length = | 0x08 | 8 octet length of Entity Identifier |
| content = | 0xfedcba9876543210 |  |
| replacingSubjectKeyIdentifier OCTET STRING: |  |  |
| tag = [UNIVERSAL 4] primitive; | 0x04 | Tag for OCTET STRING |
| length = | 0x08 | length of KeyIdentifier |
| content = | 0xABCDEABCDEABCDEA | KeyIdentifier |
| ReplacementOutcome SEQUENCE: |  |  |
| tag = [UNIVERSAL 16] constructed; | 0x30 | Tag for SEQUENCE |
| length = | 0x34 |  |
| content = |  |  |
| affectedTrustAnchorCell TrustAnchorCellIdentifier SEQUENCE: |  |  |
| tag = [UNIVERSAL 16] constructed; | 0x30 | Tag for SEQUENCE |
| length = | 0x07 |  |
| content = |  |  |
| trustAnchorCellRemotePartyRole RemotePartyRole INTEGER: |  |  |
| tag = [UNIVERSAL 2] primitive; | 0x02 | Tag for INTEGER |
| length = | 0x01 |  |
| content = | 0x02 | Value for supplier |
| trustAnchorCellKeyUsage KeyUsage BIT STRING: |  |  |
| tag = [UNIVERSAL 3] primitive; | 0x03 | Tag for BIT STRING |
| length = | 0x02 |  |
| content = | 0x0308 |  |
| statusCode StatusCode ENUMERATED: |  |  |
| tag = [UNIVERSAL 10] primitive; | 0x0A | Tag for ENUMERATED |
| length = | 0x01 |  |
| content = | 0x00 | Value for success |
| existingSubjectUniqueID OCTET STRING: |  |  |
| tag = [UNIVERSAL 4] primitive; | 0x04 | Tag for OCTET STRING |
| length = | 0x08 | 8 octet length of Entity Identifier |
| content = | 0x123456789abcdef0 |  |
| existingSubjectKeyIdentifier OCTET STRING: |  |  |
| tag = [UNIVERSAL 4] primitive; | 0x04 | Tag for OCTET STRING |
| length = | 0x08 | length of KeyIdentifier |
| content = | 0x0987654321098765 | KeyIdentifier |
| replacingSubjectUniqueID OCTET STRING: |  |  |
| tag = [UNIVERSAL 4] primitive; | 0x04 | Tag for OCTET STRING |
| length = | 0x08 | 8 octet length of Entity Identifier |
| content = | 0xfedcba9876543210 |  |
| replacingSubjectKeyIdentifier OCTET STRING: |  |  |
| tag = [UNIVERSAL 4] primitive; | 0x04 | Tag for OCTET STRING |
| length = | 0x08 | length of KeyIdentifier |
| content = | 0xFEDCBFEDCBFEDCBF | KeyIdentifier |

Table 18.3.2b: Illustrative @UpdateSecurityCredentials.ResponsePayload instantiation – DER encoding

## Amend Section 19.3 Use Cases as follows:

**19.3 Embedded Use Cases**

Table 19.3 contains the Use Cases that fulfil the interface requirements to cover Commands (and their Responses) and Alerts (where applicable). In addition, it includes ZSE Message Templates.

Note: DLMS COSEM methods that have values which have an impact on the execution of the method (that is, methods with input values that are not integer(0)), the DLMS part of the Mapping Table and the Use Case include two or more rows. One row contains the method, and the subsequent row(s) contain the value(s) to be sent with the method.

A number of Use Cases are also covered in GBCS main body. These are identifiable from the Table of Contents.



Table 19.3: Use Cases

## Amend Section 20 Mapping Table as follows:

## 20 Mapping Table

Table 20 contains the Mapping Table from which the Use Cases and Message Templates were generated. These tables map between SMETS attributes and methods, SEC Service Requests, Use Cases, DLMS COSEM attributes and methods and ZSE clusters, attributes and commands.

In addition to the Use Cases, certain columns in the Mapping Table are directly referenced from this document.

Please note that in the SMETS required objects tab only rows marked ‘E’ (External to HAN) in column F are fully specified, since those rows relate to Remote Party Messages. Other rows are only specified to the extent that these elements of Remote Party Messages rely on them.



Table 20: Mapping Table

**Amend Section 21 Glossary as follows:**

Execution Counter

Shall have the meaning defined in Section 4.3.1.5 and in Section 13.3.5.3.

1. <https://www.itu.int/rec/T-REC-X.690/en> [↑](#footnote-ref-2)
2. <https://www.itu.int/rec/T-REC-X.680/en> [↑](#footnote-ref-3)