# Appendices

## Priority Mapping (Informative)

The current version of the IEEE 802.1 standard describes the use of user priorities and access priorities in a bridged-network environment. User priorities are the priorities that a user or application requests be associated with its traffic. Access priorities are the number of differentiated traffic classes that a MAC provides. In subclause 7.7.3, IEEE 802.1D provides the following mapping of user priorities to traffic classes.

Table 13‑1: Recommended User Priority-to-Traffic Class Mappings

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | Number of Available Traffic Classes | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| User Priority | 0 (default) | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 2 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 3 | 0 | 0 | 0 | 1 | 1 | 2 | 2 | 3 |
| 4 | 0 | 1 | 1 | 2 | 2 | 3 | 3 | 4 |
| 5 | 0 | 1 | 1 | 2 | 3 | 4 | 4 | 5 |
| 6 | 0 | 1 | 2 | 3 | 4 | 5 | 5 | 6 |
| 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

***Note***: The rationale behind the choice of values in Table 13‑1 is discussed in H.2 of IEEE 802.1D. A consequence of the mapping shown is that frames carrying the default user priority (0) are given preferential treatment relative to user priorities 1 and 2 in HLEs that implement four or more traffic classes.

As with HomePlug AV, GREEN PHY provides four differentiated traffic classes at the PHY level, corresponding to the four channel access priorities. In Table 13‑1, the mapping from column four (highlighted) is recommended, where HomePlug channel access priorities 0 through 3 correspond to traffic classes 0 through 3.

Both HomePlug AV and GREEN PHY QoS functions differentiate between eight levels of user priority, eliminating the need for mapping at the higher layers.

This priority mapping allows both HomePlug AV and GREEN PHY to operate with the industry standard Request for Comments (RFC) 2205 Resource Reservation Protocol (RSVP [16]) and the Internet draft standard Subnet Bandwidth Manager (SBM) to provide differentiated QoS levels for multimedia traffic.

Table 13‑2 is derived from H.2 of IEEE 802.1D and defines the user priorities that should be assigned to application classes.

Table 13‑2: Recommended Application Class-to-User Priority Mappings

|  |  |
| --- | --- |
| User Priority | Application Class |
| 7 | a) Network Control — characterized by a “must-get-there” requirement to maintain and support the network infrastructure. |
| 6 | b) “Voice” — characterized by less than 10 ms delay, and hence maximum jitter (one-way transmission through the LAN infrastructure of a single campus). |
| 5 | c) “Video” or “Audio” — characterized by less than 100 ms delay. |
| 4 | d) Controlled Load — important business applications subject to some form of “admission control,” be it pre-planning of the network requirement at one extreme to bandwidth reservation per flow at the time the flow is started at the other. |
| 3 | e) Excellent Effort — or “CEO’s best effort,” the best-effort type services that an information-services organization would deliver to its most important customers. |
| 0 | f) Best Effort — LAN traffic as we know it today. (This user priority is actually serviced at a higher priority than user priorities 1 and 2 to accommodate legacy entities.) |
| 1,2 | g) Background — bulk transfers and other activities that are permitted on the network, but that should not impact the use of the network by other users and applications. |

## User Experiences (UEs) (Informative)

This informative section describes typical user experiences (UEs) that are supported by HomePlug AV and GREEN PHY. These user experiences reflect the way in which the user interacts with the devices as they are being configured. They will depend upon how the device manufacturer implements the user interface for the device and what configuration is performed before the user installs the device. The various UEs reflect and are supported by underlying protocols that distribute the Network Membership Key (NMK), which defines an AV Logical Network. Possession of the NMK allows a device to join the corresponding AVLN. Two NMK distribution methods (supporting UE2 and UE3) are very secure, while the third (supporting UE4) sacrifices a measure of security for convenience.

Four basic UEs are anticipated:

* User plugs devices in a set into the outlets and they connect by themselves
* User enters NPW to get a device to join an AVLN (devices with rich user interfaces)
* User enters DPW to add another device to an AVLN (at least one device with rich user interface)
* User pushes a button on each of two devices to get them to connect to each other

In addition, the manufacturer should provide each device with a reset mechanism (e.g., a reset button) and should provide the user with some indication of the state of the device and the network. If the device supports the Simple Connect mechanism (refer to Section 13.2.4) using button presses, and only has one button, then it is recommended that only a long duration button press be interpreted as “Reset.” In this case, reset should cause the device to change Security Level to SL-SC. Care should be taken to shield the button from inadvertent presses, and a button press sustained for too long should be ignored.

Indication to the user of the state of the device and the network may be done with one or more LEDs, with one or more colors. As a minimum, the device should indicate:

* That the power is on.
* Whether the device detects other traffic on the power line.
* Whether the device is part of an AVLN.
* Which security mode the device is in (i.e., SL-HS or SL-SC).

Feedback indicating an error when a user presses the button when the device is in SL-HS is also desirable.

### UE1 – Preconfigured Set of Devices

A manufacturer may package several devices as a set (e.g., a home theater audio component set). In this case, the manufacturer may choose to set the NMK on all devices in the set to the same value.

Since all the devices have the same NMK out of the box, the user need only plug them in for all the devices to discover one another and form an AVLN without user intervention.

If the Security Level is set to SL-HS, then it is recommended that the manufacturer derive the NMK from a random NPW, and that the NPW be included in the packaging. This is so that the user can add new devices to the network using UE2. UE4 will not be possible.

If the Security Level is set to SL-SC, then the user can employ Simple Connect methods (UE4) to add new devices. UE2 will not be possible.

Regardless of Security Level, if one or more of the devices has a user interface that allows the user to enter the DPW of a new device, then UE3 is also available.

This mechanism provides very high security for the devices in the pre-packaged set, as long as no other devices are added, even if the Security Level of the network is SL-SC. For networks at SL-SC, adding devices to the network using Simple Connect (UE4) exposes the network to possible compromise.

Note: This approach has the drawback that a user who purchases multiple such sets and installs them without any user intervention will create one AVLN per set of devices. This will cause proliferation of AVLNs, neighbor networks, use up Beacon slots, and create unnecessary overhead in AVLN management. The user should be advised to add new devices to an existing AVLN, even when they come as a preconfigured set.

|  |
| --- |
| Informative Text  Manufacturers are cautioned against shipping large numbers of devices with the same NPW and hence, NMK. All populations of such devices capable of communicating with one another will attempt to form a single AVLN, which will degrade performance considerably if the network is physically dispersed enough to have hidden nodes, particularly if there are multiple layers of hidden nodes. |

### UE2 – Network Password Entry

A device that has a user interface suitable for entry of alphanumeric characters can allow the user to select and enter a Network Password (NPW). If there are other devices already in an AVLN, then the user must know the NPW of the existing network to cause the new device to join the existing AVLN. The Security Level for the new device will be SL-HS, so UE4 will not be possible for devices in this network.

This approach is very secure for key distribution, since the NMK is never sent over the network. However, human selection of the NPW can expose networks formed in this manner to password-guessing attacks. It is therefore recommended that human selected passwords be much longer than the minimum length of 12 characters.

### UE3 – Device Password Entry

If an AVLN has a device with a user interface suitable for entry of alphanumeric characters, the user can enter the Device Password (DPW) of a device the user wishes to add to the AVLN. The DPW is a unique alphanumeric string set by the manufacturer and provided with the device (e.g., with the documentation and/or printed on a label attached to the device itself). The user needs only to enter the DPW in the appropriate text box to cause the device on which the DPW is entered to distribute the NMK to the device whose DPW is entered, causing it to join the existing network.

This distribution method is very secure (as long as the DPW is not easily guessable) and is supported for both security levels.

### UE4 – Simple Connect (Button Push)

To support easy connection of devices packaged separately that might not have rich user interfaces (i.e., capable of character entry) supporting UE2 or UE3, both HomePlug AV and GREEN PHY support the Simple Connect experience. Generically, the user presses a button on one device, then presses a button on another device within a short amount of time to cause these two devices to join the same network. While the amount of time a device remains promiscuous is the vendor’s choice, the recommended range for this vulnerable time is between 30 seconds and two minutes, with one minute as the default value. Under some circumstances, a vendor may elect a value outside this range, or even to provide the choice of duration to the user.

This mechanism trades off convenience for users and low-cost user interfaces for lowered security. Simultaneous execution of this mechanism can cause networks to admit stations other than those desired by the user, and sufficiently equipped and sophisticated attackers can compromise the key exchange itself, so this is only recommended for non-sensitive information applications.

While this experience is described below by the user pushing buttons, a device with richer interface may have a menu selection that is equivalent to pushing a button and may be more specific about whether a device is to keep its current NMK (“Add” the other device) or discard its current NMK (“Join” the other device). The descriptions below assume a single button that the user presses briefly to indicate the desire for two devices to join with each other in a common network.

Note that a device that has an NMK-HS shall be configured to ignore button presses, as the NMK-HS must not be distributed using UKE (the protocol underlying the Simple Connect experience). In this case, the user must first change the Security Level of the device (by changing its NMK or resetting) before the device will respond to the button presses and permit the Simple Connect experiences.

#### UE4a – Two New Devices Form a New Network Using SC

When a user has two devices that are not already part of a network, the user needs only to press the button on one device, then press the button on the other device within a reasonable amount of time. The time constraint is determined by the manufacturer, but 1-2 minutes should be typical. As long as no other devices have their buttons pushed as this process continues, the two devices should detect one another and form a new AVLN.

In the unlikely event that one or both of the devices is “recruited” by some third device (e.g., a neighbor is adding devices using Simple Connect at the same time), the user may reset the device(s).

#### UE4b – Adding a New Device to an Existing Network Using SC

When the user wishes to add a new device to an existing network, the user must press the button on the new device, and press the button on a device that is already in the desired network. The two devices should quickly detect one another and the device in the existing network should share the NMK of that network with the new device. The order in which the buttons are pressed does not matter, but they should be pressed within the time constraint(s) set by the manufacturer(s).

#### UE4c – Adding Multiple New Devices Using SC Chaining

To add several new devices to an existing network using the button-push mechanism described in Section 13.2.4, the user must press two buttons for each new device (the button on the new device, and the button on a device that is already in the desired network). A vendor may implement a mode of operation in which a device in the network remains promiscuous until some period of time elapses in which new devices are added .This allows the user to press the button once on one device in the network, then press the button on each new device once.

### Changing Security Levels on a Device

Since networks are defined by their NMKs, and NMKs are associated with a Security Level that determines how they may be distributed, the user may find that a device is using an NMK with a Security Level that is incompatible with their needs. This is the case when a device is configured to use Simple Connect, but the user desires a greater level of security. Conversely, a user may have a device that has an NMK-HS, and consequently ignores button presses (except for “Reset”). In each of these cases, the user needs to be able to change the SL of the device. It is possible to do this through a menu selection in a device with a rich user interface, but may be done in other ways also.

#### Changing SL-HS to SL-SC

When a user has a device with an NMK-HS, it must not distribute the NMK-HS using UKE (the protocol underlying Simple Connect experience). Thus the device shall ignore “Join” and “Add” button presses (and should indicate an error to the user if they are pressed). To cause the device to enter Simple Connect Mode, the user may:

1. Specify a Security Level mode change to SL-SC from the user interface.
2. Enter the device’s DPW on another device that is in SL-SC (the other device must have a rich user interface).
3. Reset the device using a “Reset” button or menu selection.

In all cases, the device will discard its old NMK and either receive a new one (case 2) or generate a new NMK (cases 1 and 3).

#### Changing SL-SC to SL-HS

When a user has a device that is in SL-SC but wishes the device to be in SL-HS, the user may:

1. Specify a Security Level mode change to SL-HS from the user interface.
2. Enter an NPW and specify SL-HS (the device must have a rich user interface).
3. Enter the device’s DPW on another device that is in SL-HS (the other device must have a rich user interface).

As when changing from SL-HS to SL-SC, the device will discard its old NMK-SC and either generate a new one (case 1) or receive the NMK-HS (cases 2 and 3). If an NMK-HS is generated, the device may do this by generating a random NPW first, from which the NMK-HS is then derived. At the option of the manufacturer, this NPW may be displayed to the user in order that it can be used in UE2 with other devices.

## Security State Transition Diagrams

This section describes the state machine for security and key management.

### State Definitions for Security Protocol State Machine

Figure 13‑1 and following provide diagrams showing security state transitions and their triggers.

In these figures, detailed states of the protocols are not shown.

An Unassociated STA may create its own AVLN due to:

* Failure to detect any Beacons
* Detection of another Unassociated STA that has a matching NID
* When in the DAK-encrypted NMK provisioning protocol, detection of another STA in SC-Join that is less CCo-capable
* When in SC-Join, detection of another STA in SC-Join that is less CCo-capable

These cases are not distinguished in the diagrams.



Figure 13‑1: State Transition Diagram for HS Security Level



Figure 13‑2: State Transition Diagram for Simple-Connect Security Level

## Test Vectors

The HomePlug AV specification includes a set of transmit test vectors that provide a variety of examples of MAC Frames and MPDUs, as well as a sample Matlab implementation of an AV transmit PHY.

These vectors are distributed as a ZIP file called AVVectors v1.1.zip. This file should be verified to be a true copy of the original by checking the Message Digest using SHA‑256 as specified in FIPS 180-2 (refer to Section 1.1).

Table 13-2b: Test Vectors

|  |  |
| --- | --- |
| Filename | SHA-256 Message Digest |
| AVVectors v1.1.zip | 437ce9d1e0c70255ee1f81aca96a6ab920a7827b177011c4c9598f8c116c9ca6 |

In the ZIP file, there is a directory called \AVVectors\PHY that contains the sample PHY model. The directory \AVVectors\Work contains the various transmit test vectors, inputs, and outputs. Also included is a readme.txt file that explains the inputs and outputs used in the vectors. For more information about the AV vectors, refer to the readme.txt file.

For reference, the MatLab vectors were run with MatLab v7.0.1 using the MatLab Communications Toolbox. All input files are in plain text format. Output files are provided in MatLab .MAT format as well as in plain text.

The test vectors and the PHY model contained in the AVVectors v1.1.zip file shall be treated as Normative.

## Example Hashed NMK, Hashed NID, and NMK Provisioning MME Using DAK

Table 13-3 shows examples of passwords hashed to AES Encryption Keys, with AES octet number 0 corresponding to the leftmost octet (byte array position 0) of the hashed key octet string. Table 13-4 shows an example NID hashed from the NMK in Table 13-3. Table 13-4 and Table 13-5 show an example MME provisioning the NMK with the DAK mechanism. Table 13-4 shows an example of a CM\_SET\_KEY.REQ message that contains the NMK from Table 13-3. Table 13-6 shows an example CM\_ENCRYPTED\_PAYLOAD.IND message where the CM\_SET\_KEY.REQ payload MME from Table 13-5 is encrypted with the DAK from Table 13-3.

Note: The “Test Value” column in Table 13-5 and Table 13-6 indicates the test values in hexadecimal format, starting from the least-significant octet. For example, the OSA field in these examples has a value of 004647484950. The least-significant and most-significant octets in the field are 0x00 and 0x50, respectively.

Table 13‑3: Example AES Encryption Keys Hashed from Passwords

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Octet | NMK-HS | DAK | Description | Description | Description |
|  | “HomePlugAV0123” | “DAK\_Password” | Key Octet Order | AES Key order for Encrypted Payload | AES Key order for PBB Encryption |
| 0 | B5 | EE | Leftmost Key Octet | AES Key [0 - 7] | AES Key [7 - 0] |
| 1 | 93 | 7F |  | AES Key [8 - 15] | AES Key [15 - 8] |
| 2 | 19 | 57 |  | AES Key [16 - 23] | AES Key [23 - 16] |
| 3 | D7 | 88 |  | AES Key [24 - 31] | AES Key [31 - 24] |
| 4 | E8 | E2 |  | AES Key [32 - 39] | AES Key [39 - 32] |
| 5 | 15 | A0 |  | AES Key [40 - 47] | AES Key [47 - 40] |
| 6 | 7B | 21 |  | AES Key [48 - 55] | AES Key [55 - 48] |
| 7 | A0 | C9 |  | AES Key [56 - 63] | AES Key [63 - 56] |
| 8 | 01 | 99 |  | AES Key [64 - 71] | AES Key [71 - 64] |
| 9 | B0 | 46 |  | AES Key [72 - 79] | AES Key [79 - 72] |
| 10 | 18 | 9A |  | AES Key [80 - 87] | AES Key [87 - 80] |
| 11 | 66 | C5 |  | AES Key [88 - 95] | AES Key [95 - 88] |
| 12 | 9C | 2A |  | AES Key [96 - 103] | AES Key [103 - 96] |
| 13 | CE | F3 |  | AES Key [104 - 111] | AES Key [111 - 104] |
| 14 | E3 | 0A |  | AES Key [112 - 119] | AES Key [119 - 112] |
| 15 | 0D | 06 | Rightmost Key Octet | AES Key [120 - 127] | AES Key [127 - 120] |

Table 13‑4: Example NID Offset Hashed from NMK-HS with Appended Security Level

|  |  |  |
| --- | --- | --- |
| Octet | NID | Description |
| 0 | 0x02 | Leftmost Octet of 52 Bit Hashed NID Offset |
| 1 | 0x6B |  |
| 2 | 0xCB |  |
| 3 | 0xA5 |  |
| 4 | 0x35 |  |
| 5 | 0x4E |  |
| 6 | 0x18 | Left Nibble = 0x1 HS Security Level  Right Nibble = 0x8, the rightmost nibble of the 52 Bit NID offset |

Table 13‑5: Example CM\_SET\_KEY.REQ Message Provisioning NMK Using the DAK

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Field Size (Octets) | Definition | Example Value  (Left Octet = LSByte) |
| ODA | 6 | Original Destination Address | 003132333435 |
| OSA | 6 | Original Source Address | 004647484950 |
| VLAN Tag | 0 or 4 | IEEE 802.1Q VLAN Tag (optional) | None |
| MTYPE | 2 | 0x88e1 (IEEE-assigned Ethertype) | 88e1 |
| MMV | 1 | Management Message Version | 01 |
| MMTYPE | 2 | Management Message Type | 0860  CM\_SET\_KEY.REQ |
| FMI | 2 | Fragmentation Management Information | 0000 |
| Key Type | 1 | Key Type | 01  NMK (AES-128) |
| My Nonce | 4 | Random number that will be used to verify next message from other end; in encrypted portion of payload. | 00112233 |
| Your Nonce | 4 | Last nonce received from recipient; it will be used by recipient to verify this message; in encrypted portion of payload. | 44332211 |
| PID | 1 | Protocol for which Set Key is asserted  Note: This is included since MME is not always in encrypted payload.  Refer to Section 11.5.2.3 for information. | 02  Provision STA with NMK using DAK |
| PRN | 2 | Protocol Run Number (refer to Section 11.5.2.4) | 2D37 |
| PMN | 1 | Protocol Message Number (refer to Section 11.5.2.5) | 03 |
| CCo Capability | 1 | The two LSBs of this field contain the STA’s CCo capability. The interpretation of these bits is the same as in Section 4.4.3.15.4.6.2. The six MSBs of this field are set to 0b000000 | 02  Level-2 CCo Capable |
| NID | 7 | Network ID of transmitting STA  The 54 LSBs of this field contain the NID (refer to Section 4.4.3.1). The two MSBs shall be set to 0b00. | 026BCBA5354E18  NID from Table 13‑4: Example NID Offset Hashed from NMK-HS with Appended Security Level |
| NewEKS | 1 | New Encryption Key Select or New Payload Encryption Key Select depending upon value of Key Type  The four LSBs of this field contain the PEKS (refer to Section 11.5.2.1 ) or EKS (refer to Section 4.4.1.5.2.8). The four MSBs shall be set to 0x0. | 01  NewEKS is ignored when Key Type is NMK |
| NewKEY | 0, 16 or 384 | New Key (none, 128-bit AES Key or 3072-bit Hash Key) | B59319D7E8157BA001B018669CCEE30D  NMK from Table 13-3 |

Table 13‑6: Example CM\_ENCRYPTED\_PAYLOAD.IND Message Provisioning NMK Using the DAK

| Field | Field Size (Octets) | Definition | Example Value  (Left Octet = LSByte) | |
| --- | --- | --- | --- | --- |
| ODA | 6 | Original Destination Address | 003132333435 | |
| OSA | 6 | Original Source Address | 004647484950 | |
| VLAN Tag | 0 or 4 | IEEE 802.1Q VLAN Tag (optional) | None | |
| MTYPE | 2 | 0x88e1 (IEEE-assigned Ethertype) | 88e1 | |
| MMV | 1 | Management Message Version | 01 | |
| MMTYPE | 2 | Management Message Type | 0660  CM\_ENCRYPTED\_PAYLOAD.IND | |
| FMI | 2 | Fragmentation Management Information | 0000 | |
| PEKS | 1 | Payload Encryption Key Select (**Unencrypted**)  The four LSBs of this field contain the PEKS. The four MSBs shall be set to 0x0. | 00  Destination STA’s DAK (AES 128 bit key) | |
| AVLN Status | 1 | AVLN status of source. (**Unencrypted**) | 05  Associated with an AVLN and PCo Capable | |
| PID | 1 | Protocol ID (**Unencrypted**) | 02  Provision STA with NMK using DAK | |
| PRN | 2 | Protocol Run Number (**Unencrypted**) | 2D37 | |
| PMN | 1 | Protocol Message Number (**Unencrypted**) | 03 | |
| IV | 16 | AES Encryption Initialization Vector (**Unencrypted**) | FEDCBA9876543210FEDCBA9876543210 | |
| Len | 2 | Length of MM, in octets (**Unencrypted**) | 3900  Length of CM\_SET\_KEY.REQ | |
| RF | 0-15 | Random Filler: A number (between 0 and 15) of random filler octets included in Encrypted Payload to make position of Protocol fields unpredictable (**Encrypted Payload**) | 2468ACE035 | **DAK used for Encryption:**  EE7F5788E2A021C999469AC52AF30A06 from Table 13‑3  **Encrypted Payload:**  A9A887CA49530931BB7360AD22B284619EAFB1078A318564A90BD1D3E629B7BD008A626840B13DBDDC3E26E512331F9E67CC44187681F653D0DBE18FBB0ED9DF095CC6B89F21657B203ED6593635663C |
| MM | Var | MM (Management Message – refer to Section 11.1) can be any legal Management Message except CM\_ENCRYPTED\_PAYLOAD.IND (**Encrypted Payload**) | See CM\_SET\_KEY.REQ above |
| CRC | 4 | Checksum on MME (**Encrypted Payload**) | F1662820 |
| PID | 1 | Protocol ID (**Encrypted Payload**) | 02  Provision STA with NMK using DAK |
| PRN | 2 | Protocol Run Number (**Encrypted Payload**) | 2D37 |
| PMN | 1 | Protocol Message Number (**Encrypted Payload**) | 03 |
| Padding | Var | To adjust size of Encrypted Payload to 128-bit boundary (**Encrypted Payload**) | ACBCD2114DAE1577C6 |
| RFLen | 1 | 0x00 – 0x0F = Length of Random Filler (Bit numbers are before encryption and after decryption). (**Encrypted Payload**)  0x10 – 0xFF = reserved | 05 |

## Example of NMK Provisioning Using UKE Mechanism

Table 13‑7: CM\_GET\_KEY.REQ Provisioning NMK Using UKE – Message 1

through Table 13‑12: CM\_ENCRYPTED\_PAYLOAD.IND Provisioning NMK Using UKE – Message 4

show example messages provisioning the NMK using the UKE mechanism.

* Table 13‑7: CM\_GET\_KEY.REQ Provisioning NMK Using UKE – Message 1
* is Message-1, **CM\_GET\_KEY.REQ,** containing the first Hash Key.
* Table 13‑8: CM\_GET\_KEY.CNF Provisioning NMK Using UKE – Message 2
* is Message-2, CM\_GET\_KEY.CNF, containing the second Hash Key and the PEKS of the TEK that is generated.
* Table 13‑9: CM\_SET\_KEY.REQ Provisioning NMK Using UKE – Payload of Message 3
* and Table 13‑10: CM\_ENCRYPTED\_PAYLOAD.IND Provisioning NMK Using UKE – Message 3
* together are an example of Message 3, the MME provisioning the NMK. Table 13‑9: CM\_SET\_KEY.REQ Provisioning NMK Using UKE – Payload of Message 3
* is the CM\_SET\_KEY.REQ MME. Table 13‑10: CM\_ENCRYPTED\_PAYLOAD.IND Provisioning NMK Using UKE – Message 3
* shows the CM\_ENCRYPTED\_PAYLOAD.IND MME that carries the CM\_SET\_KEY.REQ. The payload of the CM\_ENCRYPTED\_PAYLOAD.IND is encrypted with the TEK that was generated from the Hash Keys and whose PEKS was set by the sender of Message 2.
* Table 13‑11: CM\_SET\_KEY.CNF Provisioning NMK Using UKE – Payload of Message 4
* and Table 13‑12: CM\_ENCRYPTED\_PAYLOAD.IND Provisioning NMK Using UKE – Message 4
* together are an example of Message 4, the MME acknowledging the receipt of the NMK. Table 13‑11: CM\_SET\_KEY.CNF Provisioning NMK Using UKE – Payload of Message 4
* is the CM\_SET\_KEY.CNF MME. Table 13‑12: CM\_ENCRYPTED\_PAYLOAD.IND Provisioning NMK Using UKE – Message 4
* shows the CM\_ENCRYPTED\_PAYLOAD.IND MME that carries the CM\_SET\_KEY.CNF. The payload of the CM\_ENCRYPTED\_PAYLOAD.IND MME is encrypted with the NMK.

### CM\_GET\_KEY.REQ

Table 13‑7: CM\_GET\_KEY.REQ Provisioning NMK Using UKE – Message 1

| Field | Field Size (Octets) | Definition | Example Value  (Left Octet = LSByte) |
| --- | --- | --- | --- |
| ODA | 6 | Original Destination Address | 003132333435 |
| OSA | 6 | Original Source Address | 004647484950 |
| VLAN Tag | 0 or 4 | IEEE 802.1Q VLAN Tag (optional) | None |
| MTYPE | 2 | 0x88e1 (IEEE-assigned Ethertype) | 88e1 |
| MMV | 1 | Management Message Version | 01 |
| MMTYPE | 2 | Management Message Type | 0C60  CM\_GET\_KEY.REQ |
| FMI | 2 | Fragmentation Management Information | 0000 |
| Request Type | 1 | Request Type  0x00 = direct  0x01 = relayed  0x02 - 0xFF = reserved | 00  direct |
| Requested Key Type | 1 | Requested Key Type  Interpretation of this field is the same as in Section 11.5.4.1. | 04  Hash Key (Random-3072) |
| NID | 7 | Network ID of transmitter or NID of AVLN that transmitter wants to join.  The 54 LSBs of this field contain the NID (refer to Section 4.4.3.1). The two MSBs shall be set to 0b00. | 3F5B4FDC4D3D05 |
| My Nonce | 4 | Random number that will be used to verify next message from other end (required for all methods). | FFEEDDCC |
| PID | 1 | Protocol ID | 03  Provision STA with NMK using UKE |
| PRN | 2 | Protocol Run Number | AB34 |
| PMN | 1 | Protocol Message Number | 01 |
| HASH KEY | var | Hash Key is present only when Requested Key Type is HASH KEY ||

### CM\_GET\_KEY.CNF

Table 13‑8: CM\_GET\_KEY.CNF Provisioning NMK Using UKE – Message 2

| Field | Field Size (Octets) | Definition | Example Value  (Left Octet = LSByte) |
| --- | --- | --- | --- |
| ODA | 6 | Original Destination Address | 004647484950 |
| OSA | 6 | Original Source Address | 003132333435 |
| VLAN Tag | 0 or 4 | IEEE 802.1Q VLAN Tag (optional) | None |
| MTYPE | 2 | 0x88e1 (IEEE-assigned Ethertype) | 88e1 |
| MMV | 1 | Management Message Version | 01 |
| MMTYPE | 2 | Management Message Type | 0D60  CM\_GET\_KEY.CNF |
| FMI | 2 | Fragmentation Management Information | 0000 |
| Result | 1 | Result  0x00 = key granted  0x01 = request refused  0x02 = unsupported method/key type  0x03 - 0xFF = reserved | 00  key granted |
| KeyType | 1 | Key Type (refer to Section 11.5.4.1).  Values = NMK (AES-128) | NEK (AES-128) | TEK (AES-128 | HASH\_KEY (Random-3072) | nonce-only) are permitted in this message. | 04  Hash Key (Random-3072) |
| My Nonce | 4 | Random number that will be used to verify next message from other end; in encrypted portion of payload. | 33221100 |
| Your Nonce | 4 | Last nonce received from recipient; it will be used to by recipient to verify this message; in encrypted portion of payload. | FFEEDDCC |
| NID | 7 | Network ID of STEI STA  The 54 LSBs of this field contain the NID (refer to Section 4.4.3.1). The two security bits shall be set to 0b00. | 3F5B4FDC4D3D05 |
| EKS | 1 | EKS or PEKS value depending upon Key Type  The four LSBs of this field contain the PEKS (refer to Section 11.5.2.11) or EKS (refer to Section 4.4.1.5.2.8. The four MSBs shall be set to 0x0. If nonce-only, set to 0x0F. | 03  This is the PEKS assigned to the TEK that will generated from the Hash Keys |
| PID | 1 | Protocol ID | 03  Provision STA with NMK using UKE |
| PRN | 2 | Protocol Run Number | AB34 |
| PMN | 1 | Protocol Message Number | 02 |
| Key | var | Encryption or Hash Key ||

### TEK Computation

The TEK used below is computed as outlined in Section 7.10.2.6. The computed TEK = 36 6A 3B 2D 8A 0F C6 DD CA E8 C5 56 36 7D 4B EB.

### CM\_SET\_KEY.REQ in CM\_ENCRYPTED\_PAYLOAD.IND

Table 13‑9: CM\_SET\_KEY.REQ Provisioning NMK Using UKE – Payload of Message 3

| Field | Field Size (Octets) | Definition | | Example Value  (Left Octet = LSByte) | |
| --- | --- | --- | --- | --- | --- |
| ODA | 6 | Original Destination Address | | 003132333435 | |
| OSA | 6 | Original Source Address | | 004647484950 | |
| VLAN Tag | 0 or 4 | IEEE 802.1Q VLAN Tag (optional) | | None | |
| MTYPE | 2 | 0x88e1 (IEEE-assigned Ethertype) | | 88e1 | |
| MMV | 1 | Management Message Version | | 01 | |
| MMTYPE | 2 | Management Message Type | | 0860  CM\_SET\_KEY.REQ | |
| FMI | 2 | Fragmentation Management Information | | 0000 | |
| Key Type | 1 | Key Type | | 01  NMK (AES-128) | |
| My Nonce | 4 | Random number that will be used to verify next message from other end; in encrypted portion of payload. | | FFEEDDCC  Note: It is not necessary for a STA generate more than one new nonce within the same protocol run | |
| Your Nonce | 4 | Last nonce received from recipient; it will be used by recipient to verify this message; in encrypted portion of payload. | | 33221100 | |
| PID | 1 | Protocol for which Set Key is asserted  Note: This is included since MME not always in encrypted payload)  Refer to Section 11.5.2.3 for information. | | 03  Provision STA with NMK using UKE | |
| PRN | 2 | Protocol Run Number (refer to Section 11.5.2.4) | | AB34 | |
| PMN | 1 | Protocol Message Number (refer to Section 11.5.2.5) | | 03 | |
| CCo Capability | 1 | The two LSBs of this field contain the STA’s CCo capability. The interpretation of these bits is the same as in Section 4.4.3.15.4.6.2. The six MSBs of this field are set to 0b000000 | | 02  Level-2 CCo Capable | |
| NID | 7 | Network ID of transmitting STA  The 54 LSBs of this field contain the NID (refer to Section 3.4.3.1). The two MSBs shall be set to 0b00. | | 3F5B4FDC4D3D05 | |
| NewEKS | 1 | New Encryption Key Select or New Payload Encryption Key Select depending upon value of Key Type  The four LSBs of this field contain the PEKS (refer to Section 11.5.2.1 ) or EKS (refer to Section 4.4.1.5.2.8). The four MSBs shall be set to 0x0. | | 01  NewEKS is ignored when Key Type is NMK | |
| NewKEY | 0, 16 or 384 | | New Key (none, 128-bit AES Key or 3072-bit Hash Key) | | 0088119922AA33BB  44CC55DD66EE77FF  (NMK) |

Table 13‑10: CM\_ENCRYPTED\_PAYLOAD.IND Provisioning NMK Using UKE – Message 3

| Field | Field Size (Octets) | Definition | Example Value  (Left Octet = LSByte) | |
| --- | --- | --- | --- | --- |
| ODA | 6 | Original Destination Address | 003132333435 | |
| OSA | 6 | Original Source Address | 004647484950 | |
| VLAN Tag | 0 or 4 | IEEE 802.1Q VLAN Tag (optional) | None | |
| MTYPE | 2 | 0x88e1 (IEEE-assigned Ethertype) | 88e1 | |
| MMV | 1 | Management Message Version | 01 | |
| MMTYPE | 2 | Management Message Type | 0660  CM\_ENCRYPTED\_PAYLOAD.IND | |
| FMI | 2 | Fragmentation Management Information | 0000 | |
| PEKS | 1 | Payload Encryption Key Select (**Unencrypted**)  The four LSBs of this field contain the PEKS. The four MSBs shall be set to 0x0. | 03  This is the PEKS for the TEK passed in the CM\_GET\_KEY.CNF MME | |
| AVLN Status | 1 | AVLN status of source. (**Unencrypted**) | 08  CCo of an AVLN | |
| PID | 1 | Protocol ID (**Unencrypted**) | 03  Provision STA with NMK using UKE | |
| PRN | 2 | Protocol Run Number (**Unencrypted**) | AB34 | |
| PMN | 1 | Protocol Message Number (**Unencrypted**) | 03 | |
| IV | 16 | AES Encryption Initialization Vector (**Unencrypted**) | FEDCBA9876543210FEDCBA9876543210 | |
| Len | 2 | Length of MM, in octets (**Unencrypted**) | 3900  Length of CM\_SET\_KEY.REQ | |
| RF | 0-15 | Random Filler: A number (between 0 and 15) of random filler octets included in Encrypted Payload to make position of Protocol fields unpredictable (**Encrypted Payload**) | 123456789A | **TEK used for Encryption:**  366A3B2D8A0FC6DDCAE8C556367D4BEB  **Encrypted Payload:**  5332516FB8DCAEBD2D7BA1163409CFB114D35C9FE209269E1CDF9C44DBB161CA33A55A70120B47B0A0402C692B99E8C7B0499D1A2A475A480C6F9B0CE8FC70F5C039A9DB5A5783FE8A1AF324FFFDCCE3 |
| MM | Var | MM (Management Message – refer to Section 11.1) can be any legal Management Message except CM\_ENCRYPTED\_PAYLOAD.IND (**Encrypted Payload**) | See CM\_SET\_KEY.REQ above |
| CRC | 4 | Checksum on MME (**Encrypted Payload**) | 607F75C6 |
| PID | 1 | Protocol ID (**Encrypted Payload**) | 03  Provision STA with NMK using UKE |
| PRN | 2 | Protocol Run Number (**Encrypted Payload**) | AB34 |
| PMN | 1 | Protocol Message Number (**Encrypted Payload**) | 03 |
| Padding | Var | To adjust size of Encrypted Payload to 128-bit boundary (**Encrypted Payload**) | DBF4C91A3CDA2F169B |
| RFLen | 1 | 0x00 – 0x0F = Length of Random Filler (Bit numbers are before encryption and after decryption). (**Encrypted Payload**)  0x10 – 0xFF = reserved | 05 |

### CM\_SET\_KEY.CNF in CM\_ENCRYPTED\_PAYLOAD.IND

Table 13‑11: CM\_SET\_KEY.CNF Provisioning NMK Using UKE – Payload of Message 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | Field Size (Octets) | Definition | | Example Value  (Left Octet = LSByte) |
| ODA | 6 | Original Destination Address | 004647484950 | |
| OSA | 6 | Original Source Address | 003132333435 | |
| VLAN Tag | 0 or 4 | IEEE 802.1Q VLAN Tag (optional) | None | |
| MTYPE | 2 | 0x88e1 (IEEE-assigned Ethertype) | 88e1 | |
| MMV | 1 | Management Message Version | 01 | |
| MMTYPE | 2 | Management Message Type | 0960  CM\_SET\_KEY.CNF | |
| FMI | 2 | Fragmentation Management Information | 0000 | |
| Result | 1 | 0x00 = success  0x01 = failure  0x02 – 0xFF = reserved | 00 | |
| My Nonce | 4 | Random number that will be used to verify next message from other end; in encrypted portion of payload. | 33221100 | |
| Your Nonce | 4 | Last nonce received from recipient; it will be used to by recipient to verify this message; in encrypted portion of payload. | FFEEDDCC | |
| PID | 1 | Protocol for which Set Key is confirmed  Note: This is included since MME not always in encrypted payload)  Refer to Section 11.5.2.3 for information. | 03  Provision STA with NMK using UKE | |
| PRN | 2 | Protocol Run Number (refer to Section 11.5.2.4) | AB34 | |
| PMN | 1 | Protocol Message Number (refer to Section 11.5.2.5) | FF | |
| CCo Capability | 1 | The two LSBs of this field contain the STA’s CCo capability. The interpretation of these bits is the same as in Section 4.4.3.13.4.6.2. The six MSBs of this field are set to 0b000000. | 02  Level-2 CCo Capable | |

Table 13‑12: CM\_ENCRYPTED\_PAYLOAD.IND Provisioning NMK Using UKE – Message 4

| Field | Field Size (Octets) | Definition | Example Value  (Left Octet = LSByte) | |
| --- | --- | --- | --- | --- |
| ODA | 6 | Original Destination Address | 004647484950 | |
| OSA | 6 | Original Source Address | 003132333435 | |
| VLAN Tag | 0 or 4 | IEEE 802.1Q VLAN Tag (optional) | None | |
| MTYPE | 2 | 0x88e1 (IEEE-assigned Ethertype) | 88e1 | |
| MMV | 1 | Management Message Version | 01 | |
| MMTYPE | 2 | Management Message Type | 0660  CM\_ENCRYPTED\_PAYLOAD.IND | |
| FMI | 2 | Fragmentation Management Information | 0000 | |
| PEKS | 1 | Payload Encryption Key Select (**Unencrypted**)  The four LSBs of this field contain the PEKS. The four MSBs shall be set to 0x0. | 01  NMK known to STA (AES 128 bit key) | |
| AVLN Status | 1 | AVLN status of source. (**Unencrypted**) | 05  Associated with an AVLN and PCo Capable | |
| PID | 1 | Protocol ID (**Unencrypted**) | 03  Provision STA with NMK using UKE | |
| PRN | 2 | Protocol Run Number (**Unencrypted**) | AB34 | |
| PMN | 1 | Protocol Message Number (**Unencrypted**) | FF | |
| IV | 16 | AES Encryption Initialization Vector (**Unencrypted**) | 00112233445566778899AABBCCDDEEFF | |
| Len | 2 | Length of MM, in octets (**Unencrypted**) | 2100  Length of CM\_SET\_KEY.CNF | |
| RF | 0-15 | Random Filler: A number (between 0 and 15) of random filler octets included in Encrypted Payload to make position of Protocol fields unpredictable (**Encrypted Payload**) | 3456789012 | **NMK used for Encryption:**  0088119922AA33BB44CC55DD66EE77FF  **Encrypted Payload:**  8648E0A97FC3CE46283CE6D43F9C44E38DD55CC469216EACB8D873D7EC1A3369D38CFEC56088072550D6684B0A1C001A |
| MM | Var | MM (Management Message – refer to Section 11.1) can be any legal Management Message except CM\_ENCRYPTED\_PAYLOAD.IND (**Encrypted Payload**) | See CM\_SET\_KEY.CNF above |
| CRC | 4 | Checksum on MME (**Encrypted Payload**) | B1FBF73D |
| PID | 1 | Protocol ID (**Encrypted Payload**) | 03  Provision STA with NMK using UKE |
| PRN | 2 | Protocol Run Number (**Encrypted Payload**) | AB34 |
| PMN | 1 | Protocol Message Number (**Encrypted Payload**) | FF |
| Padding | Var | To adjust size of Encrypted Payload to 128-bit boundary (**Encrypted Payload**) | 34 |
| RFLen | 1 | 0x00 – 0x0F = Length of Random Filler (Bit numbers are before encryption and after decryption). (**Encrypted Payload**)  0x10 – 0xFF = reserved | 05 |

## Third Party Managed Security Deployment Scenarios

To use 3rd party managed security, at least one of the STAs in the network must have HLE software to communicate with the 3rd party security manager know as the Authentication Server (AS). Such a STA is called an Authenticator STA. How the Authenticator communicates with the AS is outside of scope.

There are three ways to authorize a STA (called the Supplicant STA) using a 3rd party security manager:

1. DAK
2. Push-button
3. Direct entry

### DAK Authorization

DAK authorization is recommended to be used when the network operator has knowledge of the STAs that will be part of the AVLN (e.g. they own or otherwise supply the STAs). DAK Authorization requires the network operator to know the MAC/DAK pair for all authenticated STAs.

The procedure to authorize a Supplicant STA using DAK Authorization is as follows:

When an Unassociated STA (aka the Supplicant STA) powers-on it broadcasts a CM\_UNASSOCIATED\_STA.IND MME (see 7.2.1). When an Authenticator STA receives an APCM\_UNASSOCIATED\_STA.IND, it sends the MAC address of the Supplicant STA to the AS. If the Supplicant STA is known by the AS (i.e., its MAC/DAK pair has been previously registered), then the AS will supply the Authenticator STA the DAK of the Supplicant STA. The Authenticator STA will then use the APCM\_AUTHORIZE.REQ/CNF/IND (Section 12.2.2.14/15/16) primitives to pass NMK via DAK to the Supplicant STA.

### Push Button Authorization

Push-button Authorization is recommended to be used when the consumer owns the STA and the network operator agrees to authenticate STAs based the STA's MAC address (i.e. the STAs MAC address must be pre-registered with the network operator. The DAK can be kept private to the consumer.

The procedure to authorize a Supplicant STA using Push Button Authorization is as follows:

When an Authenticator STA is placed in a SC-Add state, it waits until it receives an APCM\_SC\_JOIN.REQ primitive from the Supplicant STA. The Authenticator STA then sends the MAC address of the Supplicant STA to the AS. If this MAC address is know by the AS, the AS will reply to the Authenticator STA with an "OK to join" message. The Authenticator STA then responds with a CM\_SC\_JOIN.CNF.

### Direct Entry Authorization

Direct entry Authorization can be used when a STA is also an Authenticator STA. As the Authenticator STA can communicate directly with the AS, it can authenticate itself to the AS and request the NMK from the AS.

The procedure to authorize a Supplicant STA using Direct Entry Authorization is as follows:

When a STA that is also an Authenticator STA powers-on, it may request the AS for the NMK for the AVLN. When the NMK is received the STA will use the Direct Entry of the NMK procedure (Section 7.10.3.3) to set the NMK.

## PEV - EVSE Association (GREEN PHY)

Signal Level Attenuation Characterization (SLAC) enables a station to measure the signal level of its transmission at other stations in the network. It was designed for automotive applications where there may be multiple plug-in electric vehicles (PEV) and Electric Vehicle Supply Equipment (EVSE). The signal level from a PEV is measured at EVSEs in order to determine uniquely which PEV is plugged in to which EVSE. The process leading to this decision will be called Green PHY PEV-EVSE Association (GreenPPEA). Unique and secure GreenPPEA between a PEV and the EVSE is essential for proper charging and billing purposes. It is not sufficient, however, and higher level processes are needed to authorize the EVSE to supply power to the PEV plugged in to it, and for the correct account to be charged for the received energy (refer to ISO/IEC 15118 series).

|  |
| --- |
| Informative Text  Note: Plug-in Electric Vehicles (PEV) is a subcategory of different types of Electric Vehicles (EV). In this application, PEVs plug into an external AC or DC power source, an EVSE, for charging energy. The communications between the PEV and EVSE may be done over the cordset’s (SAE J1772) AC “Mains” or “Pilot” signal wires. For more information, see the SAE J2931, IEC61851-1 and related PEV standards. |

The basic idea behind the GreenPPEA protocol is illustrated in Figure 13-X. The signals sent by PEV (i.e., Green PHY 1 (GP1)) will reach the EVSE-1 (i.e., GP2) with minimal attenuation. To reach any other EVSE, the PEV signal needs to go through two open contacts and/or PLC pass band (RF) coupling, which significantly reduces the signal strength measured at other EVSE’s. Note that wires within the same cable, and even wires in different cables that are sufficiently close and parallel to each other, can experience non-negligible crosstalk, but with significant signal attenuation. PEV-EVSE association exploits this inherent asymmetry in signal strength at different EVSE’s to associate (match) the PEV to the EVSE to which it is physically connected through the charging cordset.

|  |
| --- |
| Informative Text  Matching means that the PEV has been able to select and associate with the EVSE, which then together establish a private network by exchanging an NID and NMK. Matched devices no longer participate in GreenPPEA, which simplifies the process for other unmatched (available) PEVs and EVSEs. |



Figure 13-3: PEV – EVSE Association

During the GreenPPEA protocol, the PEV and the EVSE (or more properly, a charging cable controlled by an EVSE, in case an EVSE has more than one such cable) proceeds through several states. For simplicity, these will be called PEV states and EVSE states.

* The start state for the EVSE is **Unoccupied**, meaning that no PEV is currently plugged in to the EVSE its cable.
* When a PEV plugs in to the EVSE, the EVSE can detect this and transitions to the **Unmatched** state, meaning that a PEV is plugged into it, but the PEV has not yet been identified reliably.
* When the GreenPPEA protocol completes successfully, the PEV plugged in to the EVSE is identified reliably and the EVSE state becomes **Matched**.

When the PEV unplugs from the EVSE, this is detected by the EVSE and its state returns to Unoccupied.

A PEV proceeds through several states, starting with Disconnected. When it is plugged into an EVSE, it also can detect this and it transitions to the Unmatched state, in which it carries out the GreenPPEA protocol. Upon successful completion of the protocol, the PEV enters the Matched state and remains there until it is unplugged.

Only an EVSE in the Unmatched state needs to participate in the GreenPPEA protocol, as an Unoccupied EVSE cannot be the EVSE with which a PEV should be matched, and a Matched EVSE has already identified the PEV plugged into it.

### PEV – EVSE Association Procedure

As the SLAC procedure starts, the PEV and EVSE decide whether or not to use Secure SLAC. If Secure SLAC is selected, then all subsequent messages that are optionally signed must be signed for the recipient to accept them, and all messages that are optionally encrypted must be encrypted. The sequence of steps in PEV – EVSE Association procedure is an example of how a procedure could look like. Trigger conditions and decisions may vary from the HLE application, such as the ISO/IEC 15118 series.

Configuration of GP stations in PEV and EVSE: Initialize PEV to never be CCo; Initialize EVSE to always be CCo.

1. SLAC: PEV obtains SLAC parameters from the EVSE. PEV sends Multi-Network Broadcast (MNBC) messages (referred to as M-Sounds, i.e., multi-network sounds) in SLAC process; PEV then optionally provides public key security credentials, followed by M-Sounds signed using public key cryptography.
2. Match Decision: Each unmatched EVSE processes the M-Sounds, verifying the public key signature if present. Each such EVSE sends the Sounding results to the PEV so that it can make the matching decision, optionally signed and including EVSE certificate.
3. Validation: PEV optionally uses Control Pilot validation of the selected EVSE.
4. Inform EVSE of Decision: PEV decides which EVSE is attached to it based on the ATTN\_PROFILE results from the M-Sounds responses (refer to ISO/IEC 15118-3) and validation results (if used), and sends to that EVSE a matching message, optionally signed.
5. Confirm Decision: Selected EVSE sends PEV a confirmation message with NID, NMK, EVSE's MAC address; message is optionally encrypted with the PEV's public key and signed by the EVSE.
6. Network Join: PEV joins matching EVSE's AVLN, using NID and NMK supplied by selected EVSE. (Refer to HPGP specification section 7.3.4.1)
7. Amplitude Map exchange: Optionally, EVSE and PEV agree on amplitude map.

|  |
| --- |
| Informative Text  After PEV and EVSE are matched and have formed an AVLN, the GreenPPEA protocol has completed successfully. Their higher layers then work together, possibly with other devices, to obtain account authorization and supply power, and to confirm power is being received by PEV. Refer to ISO/IEC 15118-3 |

#### Configuration of GP stations in PEV and EVSE

GP station in the PEV shall be configured never to become a CCo using the **APCM\_SET\_CCo.REQ** (Section 12.2.2.57). GP station that is configured to never be a CCo shall not respond to **CM\_SC\_JOIN.REQ** message from other STAs. PEV shall be configured to only support SLAC functions associated with the transmitter of M-Sounds using **APCM\_CONF\_SLAC.REQ** (Section 12.2.2.59).

GP station in the EVSE shall be configured to always be a CCo always when the GP station is coupled to the Control Pilot. EVSEs in Unoccupied state shall be configured to not support SLAC functions using **APCM\_CONF\_SLAC.REQ**.

#### SLAC

SLAC process enables EVSEs to measure and report the attenuation profile of M-Sound PPDUs transmitted by PEV. The behavior of PEV and EVSE is described in the following sections.

|  |
| --- |
| Informative Text  The messages sent during the SLAC process are all sent using multi-network broadcast (MNBC), even if the destination MAC address is unicast. This is necessary because not all EVSEs may belong to the same AVLN, and the PEV needs to reach all EVSEs with which it may be matched. Moreover, the PEV does not even join an AVLN until it is matched, since it has to send messages using MNBC regardless. All EVSEs (and other PEVs) that receive the SLAC messages will filter them according to the MAC address, the message, and their own state. For example, devices that are not EVSEs will ignore the M-Sounds sent by PEVs in this protocol. EVSEs that are unoccupied or are already matched will also discard these messages. Refer to section 5.4.3. |

##### PEV SLAC

The SLAC procedure at the PEV is as follows. Trigger conditions and decisions may vary from the HLE application, such as the ISO/IEC 15118 series.

1. PEV-HLE initiates the SLAC protocol by sending **CM\_SLAC\_PARM.REQ** message. This message requests the EVSE(s) to provide SLAC parameters to be used by the PEV. PEV-HLE will start a timer to track the reception of the corresponding **CM\_SLAC\_PARM.CNF**. If the **CM\_SLAC\_PARM.CNF** is not received before the timer expires, the PEV-HLE may reinitiate the SLAC protocol. The fields in the **CM\_SLAC\_PARM.REQ** shall be set as follows,

* APPLICATION\_TYPE shall be set to 0x00 (i.e., Green PHY PEV-EVSE Association).
* By default, HLE shall set the SECURITY\_TYPE to 0x00. If the PEV supports Security, SECURITY\_TYPE shall be set to 0x01 (i.e., Public Key Signature) and the message shall include Cipher Suites that are supported by the PEV.

1. PEV Green PHY station shall transmit **CM\_SLAC\_PARM.REQ** message as a multi-network broadcast.
2. When the PEV Green PHY station receives a **CM\_SLAC\_PARM.CNF** message(s), it shall pass it to the PEV-HLE.
3. PEV-HLE may receive **CM\_SLAC\_PARM.CNF** from multiple EVSEs. PEV-HLE may use one or more of the received **CM\_SLAC\_PARM.CNF** messages for determining the SLAC parameters.
4. When the **CM\_SLAC\_PARM.CNF** indicates that Secure SLAC is required, PEV-HLE shall send a **CM\_PKCS\_CERT.IND** message. The Target MAC address for this message shall be set to MAC address of the PEV Green PHY station. To ensure reliable reception of this message at all EVSEs, it is recommended that this message be transmitted at least three times by the PEV-HLE. If the **CM\_PKCS\_CERT.IND** message is larger than 502 Octets, the message shall be fragmented by the HLE (refer to Section 11.1.7).
5. PEV Green PHY station shall transmit **CM\_PKCS\_CERT.IND** message from the HLE using Multi-Network Broadcast.
6. To enable reliable reception of **CM\_START\_ATTEN\_CHAR.IND** at all EVSEs, the PEV-HLE shall send at least three **CM\_START\_ATTEN\_CHAR.IND** messages, with the fields reflecting the values of the **CM\_SLAC\_PARM.CNF** message it received.If Secure SLAC has been selected, then these messages shall be signed. PEV Green PHY station shall transmit **CM\_START\_ATTEN\_CHAR.IND** messages from the HLE using Multi-Network Broadcast.
7. The PEV-HLE shall send **CM\_MNBC\_SOUND.IND** messages. The number of **CM\_MNBC\_SOUND.IND** messages shall be based on the corresponding NUM\_SOUNDS parameter in **CM\_SLAC\_PARM.CNF**. Each **CM\_MNBC\_SOUND.IND** message is sent with additional information, including a countdown counter (indicating the number of Sounds remaining to be sent). If Secure SLAC has been selected, then the M-Sound shall be signed by the PEV. If the **CM\_MNBC\_SOUND.IND** message is larger than 502 Octets, the message shall be fragmented by the HLE (refer Section 11.1.7). PEV Green PHY station shall transmit **CM\_MNBC\_SOUND.IND** messages using Multi-Network Broadcast.

##### EVSE SLAC

The SLAC procedure at the EVSE is as follows. Trigger conditions and decisions may vary from the HLE application, such as the ISO/IEC 15118 series.

1. When an Unoccupied EVSE-HLE enters the Unmatched state ,it configures the EVSE GP to support only SLAC functions associated with the receiver of M-Sounds using **APCM\_CONF\_SLAC.REQ**.
2. EVSE Green PHY station shall pass any received **CM\_SLAC\_PARM.REQ** messages to the EVSE-HLE
3. Reception of a **CM\_SLAC\_PARM.REQ** message at an Unmatched EVSE-HLE shall cause the HLE to transmit a **CM\_SLAC\_PARM.CNF** message to the PEV. The fields in the **CM\_SLAC\_PARM.CNF** shall be set as follows,

* APPLICATION\_TYPE shall be set to 0x00 (i.e., Green PHY PEV-EVSE Association).
* By default, HLE shall set the SECURITY\_TYPE to 0x00. If both EVSE and PEV support Security, SECURITY\_TYPE shall be set to 0x01 (i.e., Public Key Signature).
* The RESP\_TYPE shall be set to 0x01 and the FORWARDING\_STA shall be set to the MAC address of the PEV.
* All other fields shall be set based on the local configuration of EVSE-HLE.

1. The Green PHY station in the EVSE shall transmit **CM\_SLAC\_PARM.CNF** message sent by the HLE to PEV using multi-network broadcast.
2. If the EVSE-HLE supports Secure SLAC, then reception of the PEV Public Key shall cause an Unmatched EVSE-HLE to store the associated Public Key for verifying the M-Sound signatures.
3. The Green PHY station in the EVSE shall pass any **CM\_START\_ATTEN\_CHAR.IND** messages to the EVSE-HLE.
4. Reception of **CM\_START\_ATTEN\_CHAR.IND** message at the EVSE-HLE shall cause EVSE-HLE to start a timer based on the Time\_Out field in the message. Expiration of this timer is used as a trigger for generating **CM\_ATTN\_CHAR.IND**. However, if Secure SLAC has been selected, then the **CM\_START\_ATTEN\_CHAR.IND** message must be signed and the signature verified, or else it is ignored.
5. The Green PHY station in the EVSE shall pass any M-Sound messages to the EVSE-HLE along with the Attenuation Profile associated with that message. Signal processing is performed by the PHY, and if the MAC is configured to receive M-Sounds, it passes both the M-Sound message and the Attenuation Profile to the HLE.
6. Reception of an M-Sound at the EVSE-HLE shall cause it to process the Attenuation Profile information. The RunID shall match the RunID sent by that PEV in the **CM\_START\_ATTEN\_CHAR.IND** message or else the M-Sound is discarded. If the process is Secure SLAC, EVSE-HLE shall only process M-Sounds with valid public key signatures. The EVSE-HLE shall also track the value Countdown Counter of any PEV whose M-Sounds it receives. For a run (i.e., based on RunID) of the SLAC protocol, this value must be monotonically decreasing, so any M-Sound from a PEV whose Countdown Counter’s Cnt value is not less than the smallest value of Cnt received from the PEV shall be discarded.
7. If the last M-Sound is received (Cnt=0) or if the timer has expired, the EVSE-HLE shall trigger the generation of **CM\_ATTN\_CHAR.IND**. This message shall be sent to the PEV, signed if the process is secure. If signed, the EVSE shall first send its public key certificate including its EVSE\_ID and attributes that identify it as an EVSE, signed by a certificate authority allowed to identify EVSEs.
8. If Secure SLAC is used, EVSE-HLE shall send **CM\_PKCS\_CERT.IND** message with its Public Certificate. The Target MAC address for this message shall be set to MAC address of the EVSE Green PHY station. To ensure reliable reception of this message by the PEV, it is recommended that this message be retransmitted until a corresponding **CM\_PKCS\_CERT.RSP** message is received by the PEV-HLE. If the **CM\_PKCS\_CERT.IND** message is larger than 502 Octets, the message shall be fragmented (refer Section 11.1.7).
9. EVSE Green PHY station shall transmit **CM\_PKCS\_CERT.IND** message from the HLE using Multi-Network Broadcast.
10. EVSE-HLE shall send the **CM\_ATTN\_CHAR.IND** message. If secure SLAC is used, this message shall be signed by the EVSE-HLE using its public key certificate. To ensure reliable reception of this message by the PEV, it is recommended that this message be retransmitted until a corresponding **CM\_ATTN\_CHAR.RSP** message is received by the PEV-HLE.
11. EVSE Green PHY station shall transmit **CM\_ ATTN\_CHAR.IND** message from the HLE using Multi-Network Broadcast.

#### Matching Decision at PEV

Matching decision enables the PEV and EVSE to determine which EVSE and PEV they are connected through the charging cordset. The matching decision is the most important part of the Green PHY PEV-EVSE association procedure. The matching decision is made by the PEV based on the **CM\_ATTN\_CHAR.IND** message that is generated by the EVSE-HLE’s as a result of SLAC (refer Section 13.8.1.4.1). These messages shall be signed if the process is secure.

The Matching decision is made by the PEV, with optional validation by the selected EVSE. The process at the PEV is as follows:

1. When the timeout as advertised in the **CM\_START\_ATTEN\_CHAR.IND** message expires, the PEV-HLE shall start a timer for receiving results from the EVSEs.
2. When the PEV Green PHY station receives a **CM\_PKCS\_CERT.IND** message, it shall pass it to the PEV-HLE.
3. When the PEV HLE receives a **CM\_PKCS\_CERT.IND** message, if Secure SLAC has been selected it shall validate the certificate as described in Section 9 of ZigBee-11185. If valid, and if the attributes and root certificate authority identify the sender as an EVSE, it shall cache the certificate for use in verifying messages signed by that EVSE.
4. When the PEV Green PHY station receives a **CM\_ATTEN\_CHAR.IND** message, it shall pass it to the PEV-HLE.
5. When the PEV-HLE receives a **CM\_ATTEN\_CHAR.IND** message it shall verify that the RunID matches that of the **CM\_START\_ATTEN\_CHAR.IND** message and store it until it has decided which EVSE is the most likely match for it. If the process is secure, it shall verify the signature on the message and discard it if it is not verified to be from an authorized EVSE.
6. When the timer for receiving results from EVSEs expires, the PEV shall consider the ATTN\_PROFILE measured at the valid reporting EVSE’s and select an EVSE according to ISO/IEC 15118-3.

#### Validation

When the PEV’s HLE determines that association Validation is required, the PEV causes a signal to be sent out-of-band and challenge the EVSE to respond appropriately. Details of the Validation procedure are described in ISO/IEC 15118-3. To facilitate the Validation protocol, Green PHY specification includes two management messages,

1. CM\_VALIDATE.REQ (refer Section 11.5.55)
2. CM\_VALIDATE.CNF (refer Section 11.5.56)

Green PHY stations only provide transport service to these CM\_VALIDATE.REQ/CNF messages. Green PHY stations do not interpret or modify the contents of these messages.

The required behavior of PEV Green PHY station is as follows,

* PEV Green PHY station shall transmit **CM\_VALIDATE.REQ** message from the HLE using Multi-Network Broadcast
* When the PEV Green PHY station receives a **CM\_VALIDATE.CNF** message, it shall pass it to the PEV-HLE

The required behavior of EVSE Green PHY station is as follows,

* When the PEV Green PHY station receives a **CM\_VALIDATE.REQ** message, it shall pass it to the PEV-HLE
* EVSE Green PHY station shall transmit **CM\_VALIDATE.CNF** message from the HLE using Multi-Network Broadcast

#### Inform EVSE of Decision

The PEV-HLE shall send the selected EVSE (if any) its decision in a **CM\_SLAC\_MATCH.REQ** message and request the NID and NMK from the selected EVSE in the **CM\_SLAC\_MATCH.CNF** reply. If the process is secure, this message is signed by the PEV-HLE. The message shall be resent at least once if a corresponding confirmation message is not received from the selected EVSE. The PEV Green PHY station shall transmit **CM\_SLAC\_MATCH.REQ** message from the HLE using Multi-Network broadcast.

The selected EVSE’s HLE shall verify the RunID as well as the signature on the message if the process is secure, and may request validation before it confirms the matching decision.

|  |
| --- |
| Informative Text  The messages sent up until the PEV and EVSE form an AVLN are all sent using multi-network broadcast (MNBC), even if the destination MAC address is unicast. This is necessary because the PEV and the EVSE have not yet joined the same AVLN. Stations that receive the MNBC messages will filter them using the MAC address. |

#### Matching Confirm by EVSE

Once the EVSE-HLE has received the match request from a PEV and optionally has validated that the PEV is connected to it using the charging cordset based procedure described in previous sections, it confirms the decision to the PEV. It also transitions to the Matched state.

The behavior of the EVSE is as follows.

1. EVSE-HLE shall send a **CM\_SLAC\_MATCH.CNF** message that includes an NID and NMK. If Secure SLAC is used, the NID and NMK are newly generated by the EVSE-HLE and this message shall be signed by the EVSE-HLE using its public certificate and encrypted using the PEV’s public key.
2. EVSE Green PHY station shall transmit **CM\_SLAC\_MATCH.CNF** message from the HLE using Multi-Network Broadcast.
3. EVSE-HLE shall configure the EVSE Green PHY station with the same NMK and default NID that it sent to PEV using the **APCM\_SEK\_KEY.REQ** primitive. This causes the EVSE to become an unassociated STA waiting for another STA that has the same NMK/NID to join it to form an AVLN.

The PEV and EVSE will then form an AVLN using the matching NID/NMK (refer to Section 7.3.4.1). EVSE-HLE will discard any SLAC requests from other PEVs as long as the PEV remains associated.

The behavior of the PEV is as follows,

1. If Secure SLAC has been selected, the Green PHY station at the PEV passes any received **CM\_PKCS\_CERT.IND** to its HLE
2. Reception of **CM\_PKCS\_CERT.IND** causes the PEV-HLE to validate and to store the Certificate.
3. The Green PHY station at the PEV passes any received **CM\_SLAC\_MATCH.CNF** to its HLE.
4. Reception of **CM\_SLAC\_MATCH.CNF** causes the PEV-HLE to verify the RunID and the Public Key Signature (if any) of the message. Further, the PEV shall verify that the public key used for signing the **CM\_SLAC\_MATCH.CNF** is the same as the public key used for signing the **CM\_ATTN\_CHAR.IND** message from that particular EVSE. If the Public Key Signature is verified or the process is not secure, the PEV-HLE accepts the NID and NMK included in the message, causing it to leave its current network and to join the specified network, entering the Matched PEV state.

#### Network Join

When the PEV-HLE accepts an NID and NMK from a **CM\_SLAC\_MATCH.CNF**, it shall configure the PEV Green PHY station with the same NMK and default NID that it received from EVSE in **CM\_SLAC\_MATCH.CNF** using the **APCM\_SEK\_KEY.REQ** primitive. This causes the PEV to become an unassociated STA waiting for another STA that has the same NMK/NID to join it to form an AVLN. Likewise, the EVSE-HLE shall configure the EVSE Green PHY station with the same NMK and default NID that it sent to PEV using the **APCM\_SEK\_KEY.REQ** primitive. This causes the EVSE to become an unassociated STA waiting for another STA that has the same NMK/NID to join it to form an AVLN. The PEV and EVSE form an AVLN using the matching NID/NMK (refer to Section 7.3.4.1)

#### Amplitude Map exchange

EVSE may optionally provide PEV with its Amplitude Map using **CM\_AMP\_MAP.REQ** message. PEV may also optionally provide EVSE with its Amplitude Map using **CM\_AMP\_MAP.REQ**. The transmit power on various carriers used by both Green PHY stations shall be based the minimum of the EVSE’s amplitude map and the amplitude map it received from the associated PEV. HLE may configure the Green PHY station with a specific Amplitude Map using **CM\_AMP\_MAP.REQ** message. The amplitude map of a PEV and EVSE shall be reset to their default amplitude map after the PEV is disconnected from the EVSE.

|  |
| --- |
| Informative Text  The purpose of this feature is to allow the PEV to have shared control of the amplitude map, rather than it being controlled solely by the CCo (EVSE). It recognizes an uncertainty that the PEV may be sensitive to specific PLC carriers or may generate noise that could interfere with the PLC network. The extensive use of reduced carrier amplitude could reduce the bandwidth supported by the network. In addition, the amplitude map is used to cause the stations to conform to the regulatory environment in which they operate. |

### Secure SLAC Overview

Secure SLAC is an optional feature described in Section 13.8.1 that provides enhanced support from security attacks. This section addresses the following limitations of basic SLAC and conversely, the value of using secured SLAC,

* There is no mechanism to validate that M-Sounds are transmitted by the PEV. Malicious PEV’s may generate M-Sounds with strong signal that appear to be sent from a legitimate PEV. This can cause an EVSE to record false signal level attenuation values from the legitimate PEV, and cause the legitimate PEV to make the wrong decision and the EVSE possibly to provide the malicious PEV with power.
* There is no mechanism for EVSE to securely communicate its SLAC values to the PEV. Thus a rogue PEV can send the legitimate PEV misleading values that will cause it to make the wrong decision and the EVSE possibly to provide the malicious PEV with power.
* There is no mechanism for the PEV to securely communicate its decision to the selected EVSE. A rogue PEV can send a spoofed message to the EVSE to which it is connected and cause it to provide it with power under the belief that it is providing the legitimate PEV with power.

Secure SLAC uses public key signatures on the M-Sounds to prevent an EVSE from accepting spoofed M-Sounds. Verifying the RunID in each M-Sound, which must match the RunID given in the **CM\_START\_ATTEN\_CHAR.IND** message, prevents a rogue PEV from replaying M-Sounds from the legitimate PEV. Thus the EVSE’s will obtain reliable attenuation readings from a legitimate PEV. Neither can a rogue PEV spoof or replay the message from an EVSE containing SLAC data, since these messages are signed by the EVSE and must have a matching RunID, so the legitimate PEV has uncorrupted data on which to base its decision. Through public key signatures and the RunID verification, Secure SLAC also prevents a rogue PEV from spoofing or replaying the message to its EVSE telling it that it has been selected for a legitimate PEV.

#### Public Key Certificates

Public Key Certificates shall have the format as specified in ZigBee-11185. Public key certificates shall be passed as signed CMS messages (refer to RFC 5652) with null content. All received public key certificates shall be validated as described in Section 9 of ZigBee-11185r05.

The Public Key Certificate for an EVSE shall have an attribute that identifies that certificate holder as an EVSE. Similarly, the Public Key Certificate for a PEV shall have an attribute that identifies that certificate holder as a PEV.

CMS shall also be used to sign and to encrypt messages as needed using the enveloped data, and the signed data content types.

All CMS messages shall be sent using Distinguished Encoding Rules (DERs) wherever applicable.

All PEVs and EVSEs that support secure SLAC shall support at least DSA with SHA-256, RSA, and AES-128 in CBC mode. ECDSA, ECDH, and AES-256 are recommended. Refer to RFC 3565, RFC 5280, RFC 5480 and RFC 5758.

### Exchange of User Data

**CM\_SLAC\_USER\_DATA.REQ** and **CM\_SLAC\_USER\_DATA.CNF** messages may be used by PEVs and EVSEs to exchange user data as part of SLAC. For example, these messages to be used to shut the charger down or recognize the car has left (e.g. immediate disassociate). The format of these messages is described in Section 11.5.59 and Section 11.5.60.

### EVSE Power Save Restrictions

An EVSE with an associated PEV (refer to Section 13.8.1) shall not enter Power Save mode if the PEV is not in Power Save mode.