4. Acronyms and Abbreviations

ADPDU	Advertisement Protocol Data Unit
ADREQPDU	Advertisement Request Protocol Data Unit

9.11 EAPOL-Advertisement

The Packet Body of each EAPOL PDU with a packet type of EAPOL-Advertisement conveys an ADPDU. The definition of the use of the parameters is specified in Clause xx, this clause specifies their encoding.

The encoding, validation, and decoding of each ADPDU is consistent with the general rules for EAPOL PDUs. Each ADPDU (Figure xx) comprises an advertisement version followed by a number of TLVs (Type Length Values).

<table>
<tr><td colspan="2">Protocol Version</td><td></td></tr>
<tr><td colspan="2">Packet Type = EAPOL-Advertisement</td><td></td></tr>
<tr><td colspan="2">Packet Body Length</td><td>Size</td></tr>
<tr><td rowspan="3">Packet Body
(ADPDU)</td><td>Advertisement Version</td><td>1 Octet</td></tr>
<tr><td>Global TLVs</td><td>Optional,
Variable</td></tr>
<tr><td>NID Entries</td><td>Optional,
Variable</td></tr>
</table>

The standard specifies and advertisement version of 0. The TLVs consist of an optional group of Global TLVs followed by a series of NID entries. Any unknown TLVs may be ignored by the receiver.

<table>
<tr><td rowspan="2">NID Entry</td><td>NID TLV</td><td>1 TLV</td></tr>
<tr><td>Additional TLVs</td><td>0 or more</td></tr>
</table>

An ADPDU shall have at least one NID entry. Each NID entry is delineated by a NID TLV. A NID entry consists of a NID TLV followed by a series of optional TLVs. A NID TLV may not appear in the Global TLV section.

9.11.1 TLV encoding

The TLVs following the message type are identical in format to the 802.1AB TLVs as follows:

TLV Type (7 bits)	TLV information string length (9 bits)	TLV information String (0 <= n <= 511)

While the TLV format is the same as 802.1AB the type space is distinct, however it is desirable to be able to reuse TLVs from 802.1AB in an ADPDU and vice versa. Implementations of this specification only need to understand TLVs defined in this document. The order of the TLVs is important since they may be grouped together within the context of a message. Vendor specific TLVs may be defined using the Organizationally specific TLV (127) which is identical to the Organizationally Specific TLV defined in 802.1AB section 9.6. The following TLVs are defined in this specification

TLV type	TLV Name
0-123	reserved
124	Ciphersuites
125	Key Management Domain
126	NID-TLV
127	Organizationally Specific TLV

9.11.1.1 NID TLV

The network Identity TLV contains basic information about a network such as its name and its authorization procedure requirements.

NID TLV (7 bits)	TLV Information String length (9 bits)	NID Authz Mech Count (1byte)	NID Authz Mech(s) (1..n 1 byte each))	NID Name (0 <=n <=255)

The NID name is a UTF-8 encoded string used to identify a network profile supported by this authenticator. It is intended to be globally unique. The maximum length for a NID name is 255 bytes.

There is a list of authorization mechanisms associated with a NID. Each entry in the list is 1 byte long consisting of a mechanism ID (MechID) and a fallback indicator. The format of a list entry is given in table xx.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Mod	resv	resv	MechID	MechID	MechID	MechID	MechID

The mechanism ID is 5 bits (bits 0-4). Bit 7 is set to indicate if the mechanism is a fallback mechanism available if the authorization processes described in this document fail. Bit 5 and 6 are reserved. They are set to zero (0) by the sender and is ignored on receive.

Authorization Mechanism IDs are listed in table yy.

Authz Mech ID	Auth Mech
0	Open access
1	EAP
2	MKA
3	EAP + MKA
4	MKA + MACSEC
5	EAP + MKA + MACSEC
6	Higher Layer Authorization
7	Restricted Access
8	Vendor Specific
7-255	Reserved

If Authz-Mech-ID = 8 is used (vendor Specific), the inclusion of an Organizationally Specific TLV to provide information on the specific mechanism is required in the NID entry.

9.11.1.2 Organizationally Specific TLV

Organizationally specific TLVs are defined as in 802.11AB.

9.11.1.3 Ciphersuite TLV

Ciphersuite TLV (7 bits)	TLV information string length	Ciphersuite Count (1 byte)	Ciphersuite (1..n 8 bytes

	(9 bits)		each)

When the ciphersuite TLV is included in a NID entry in a EAPOL-AD message it contains a list of 802.1AE ciphersuites supported by the NID.

9.11.1.4 Key Management Domain TLV

The key management domain TLV is associated with a NID entry to provide information about the transmitting authenticator's key management domain. This is useful in identifying which cached keys are usable in a particular location.

Key Management domain TLV (7 bits)	TLV information string length (9 bits)	Key Management Domain String (0 <= n <= 255)

The key management domain TLV information string contains a string of UTF-8 characters up to 255 bytes in length. Ports that share the same Key Management Domain TLV can be assumed to share the same key cache.

9.11 EAPOL-Ad-Req

The Packet Body of each EAPOL PDU with a packet type of EAPOL-AD-REQ conveys an ADREQPDU. The definition of the use of the parameters is specified in Clause xx, this clause specifies their encoding.

The encoding, validation, and decoding of each ADPDU is consistent with the general rules for EAPOL PDUs. Each ADPDU (Figure xx) comprises an advertisement version.

Protocol Version		
Packet Type = EAPOL-Advertisment		
Packet Body Length		Size
Packet Body (ADPDU)	Advertisement Version	1 Octet

Consistent with the protocol versioning rules (9.5), EAPOL PDUs with this Packet Type are processed as normal even if they contain a Packet Body. Both the contents of the Packet Body Length field, and the contents of any Packet Body or subsequent octets are ignored.

11 EAPOL Advertisement protocol

The EAPOL advertisement protocol consists of two PDU types. The ADREQPDU is used by a supplicant to request an advertisement from the network. The ADPDU is used to deliver the advertised network capabilities to the supplicant.

The ADREQPDU may be sent to any valid unicast or multicast address defined in section 9.1.1. The supplicant should use the highest supported advertisement version that the Supplicant supports. The ADREQPDU may be sent at any time. Upon attaching to a port the supplicant should wait a short amount of time before sending an ADREQPDU to listen for unsolicited multicast advertisements from the authenticator. Upon receiving an ADREQPDU an authenticator should respond with an ADPDU within a short amount of time (how much?).

The ADPDU may be sent to any valid unicast or multicast address defined in section 9.1.1. It may be sent out unsolicited when the authenticator first becomes aware of a supplicant and periodically thereafter. If no ADREQPDUs have been received then the authenticator should send the highest supported advertisement version. If an ADREQPDU is sent by a supplicant that only supports a lower version the authenticator should send a response of that version to the supplicant. This response may be sent unicast or multicast. The processing of an ADREQPDU or the sending of an ADPDU should not create state on the authenticator.

Upon receiving the network advertisement from the authenticator the supplicant should parse the NID entries to determine what network capabilities are available.

11.1 NID Entry processing

Each NID entry begins with a NID TLV. Within the NID TLV is a network ID and a list of authorization mechanisms. The rest of the NID TLV is comprised of a series of optional TLVs. The next NID entry begins with the next NID TLV. If an unrecognized TLV is encountered then that TLV should be ignored.

The network ID is a string used to identify a network. This name is useful in aiding the supplicant to choose the right identity and credentials to use when authenticated to a NID.

The list of authorization mechanisms describes what authorization services are available. Any NID may support a combination of authorization mechanisms. A mechanism may be marked as “fallback” when a mechanism is mark as fallback then at least one other mechanism not marked as fallback must be attempted before a fallback choice is available. At least one mechanism shall not be marked fallback for a particular NID. It is not required that a NID contain a fallback mechanism. The since “fallback” indicates what is available when the processes described in this document fail the following authorization mechanisms should not be listed as fallback:

EAP

MKA
EAP + MKA
MKA + MACSEC
EAP + MKA + MACSEC

In addition since it is counter to security goals Open Access should not be marked as fallback.

The following authorization mechanisms are defined in this specification

Open Access

A network that provides open access to anyone. Open Access authorization method shall not be marked fallback.

EAP

A network supporting EAP (EAPOL-EAP) authorization defined in this document. EAP authorization mechanism shall not be marked fallback.

MKA

A network supporting MACSEC Key agreement defined in this document. The key Management domain TLV may be included to provide the supplicant with an indication of the scope of the key cache for MKA keys. MKA authorization mechanism shall not be marked fallback.

EAP + MKA

A network supporting EAP followed by MKA as defined in this specification. The key Management domain TLV may be included to provide the supplicant with an indication of the scope of the key cache for MKA keys. EAP+MKA authorization mechanism shall not be marked fallback.

MKA + MACSEC

A network supporting MKA and MACSEC as defined in this specification and 802.1AE. An option TLV may be included in the NID entry to indicate which MACSEC ciphersuites are supported. The key Management domain TLV may be included to provide the supplicant with an indication of the scope of the key cache for MKA keys. MKA+MACSEC authorization mechanism shall not be marked fallback.

EAP + MKA + MACSEC

A network supporting EAP (EAPOL-EAP) followed MKA and MACSEC defined in this specification and 802.1AE. An option TLV may be included in the NID entry to indicate which MACSEC ciphersuites are supported. The key Management domain TLV may be included to provide the supplicant with an indication of the scope of the key cache for MKA keys. EAP+MKA+MACSEC authorization mechanism shall not be marked fallback.

Higher layer authorization

A mechanism that challenge the user using a higher layer protocol such as web authentication.

Restricted Access

An open network that only provides access to a constrained set of services. Restricted access may not be marked required.

Vendor Specific

A network supporting vendor specific authorization mechanisms. An organizationally specific TLV should be included in the NID TLV to provide more information.

Examples

In these examples "supported" means "fallback" is not set.

1. University or Enterprise scenario. To get full access, user can do Higher layer authorization, EAP, EAP + MKA, or EAP + MKA + MACSEC. However, it is required that the user first attempt 802.1X before doing higher layer auth.

   This is indicated by setting, EAP, EAP + MKA, or EAP + MKA + MACSEC . to "Supported", and Restricted Access and Higher layers "Fallback",.

2. Infrastructure scenario. To get full access, device can do MKA + MACSEC. If this fails auth, they get Restricted Access. EAP, EAP + MKA or EAP + MAC + MACSEC is not supported.

   This is indicated by setting MKA + MACSEC to "Supported", and Restricted Access to "Fallback".

3. Hotspot scenario. To get full access, user can do Higher layer or EAP. If authorization fails, the user can get Restricted Access.

This is indicated by setting Higher Layer and EAP to “Supported” and Restricted Access to “Fallback”.