IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Establishing frame anonymization parameter sets text for 11bi | | | | |
| Date: 2024-07-16 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Philip Hawkes | Qualcomm |  |  | phawkes@qti.qualcomm.com |
| Duncan Ho |  |  |  |
| Jouni Malinen |  |  |  |
| George Cherian |  |  |  |

Abstract

We propose draft specification text for 10.71.3 (Establishing frame anonymization parameter sets) in TGbi draft D0.4.

Revisions:

* Rev 0: Initial version of the document.

**High level summary of the changes:**

The text describes the provides additional details when frame anonymization is enabled.

**Proposed spec text:**

The baseline for this text is Draft P802.11bi\_D0.4.

***TGbi editor: Apply the following changes to 10.71.3 (Establishing frame anonymization parameter sets)***

* Establishing frame anonymization parameter sets

This subclause describes how an AP MLD and associated non-AP MLD establish the FA parameter set for each EDP epoch for the non-AP MLD.

The non-AP MLD and AP MLD establish(#Ed) the EDP epochs used for frame anonymization as described in 10.71.2 (EDP epoch operation)(#Ed).

* The generation of EDP\_STA\_MAC values is defined in 10.7.3.1 (Generating EDP\_STA\_MAC).
* The generation of the set of EDP\_SN\_offset values is defined in 10.7.3.2 (Generating EDP\_SN\_offset).
* The generation of the set of EDP\_PN\_offset values is defined in 10.7.3.3 (Generating EDP\_PN\_offset).

#### Generating EDP\_STA\_MAC

For a given EDP Epoch, and a given Link ID the value of EDP\_STA\_MAC for the corresponding setup link is generated according to the following algorithm:

EDP\_STA\_MAC( Link ID ) ← KDF-*Hash-Length*( KDK, “EDP\_STA\_MAC\_block”, Group ID || GTn || Link ID )

where:

KDF-*Hash-Length* is the key derivation function as defined in 12.7.1.6.2 (Key derivation function (KDF)) using the hash algorithm identified by the AKM suite selector (see Table 9-190 (AKM suite selectors))

KDK is the Key Derivation Key

GTn is the reference start time of the EDP Epoch (see 9.4.2.337 (Enhanced Data Privacy (EDP) element) )

Group ID is the identifier of the group EDP Epoch (see 9.4.2.339 (Enhanced Group Privacy Availability (EGPA) element)

*Length* is the total number of bits to derive and is equal to ( *MAC\_addr\_size* − 2)

EDP\_STA\_MAC( Link ID ) is the value of EDP\_STA\_MAC used to identify the Affiliated STA of the non-AP MLD on the link identified by Link ID

Link ID identifies a link

*MAC\_addr\_size* is the number of bits in a MAC address and is equal to 48

#### Generating EDP\_SN\_offset

For a given EDP Epoch and a given sequence number space supported by the association between the non-AP MLD and AP MLD, the set of value(s) of EDP\_SN\_offset transmitted by the non-AP MLD and AP MLD are generated according to the following algorithm:

EDP\_SN\_offset\_block← KDF-*Hash-Length*( KDK, “EDP\_SN\_offset\_block”, *sns\_id,* GTn)

*start* ← 0

*finish* ← ( *ctr\_size* − 1 )

**do** *tx* in {“non-AP MLD”, “AP MLD”}

**do** *ctr\_index* = 0 to ( *ctr\_num* − 1 )

EDP\_SN\_offset(*tx*, *sns\_index*, *ctr\_index*) ← EDP\_SN\_offset\_block[ *finish* : *start* ]

*start* ← ( *start* + *ctr\_size* )

*finish* ← ( *finish* + *ctr\_size* )

**od**

**od**

where:

EDP\_SN\_offset\_block is the KDF output subsequently partitioned into the values of EDP\_SN\_offset for the counters in a given sequence number space for both the non-AP MLD and AP MLD

KDF-*Hash-Length* is the key derivation function as defined in 12.7.1.6.2 (Key derivation function (KDF)) using the hash algorithm identified by the AKM suite selector (see Table 9-190 (AKM suite selectors))

KDK is the Key Derivation Key

*sns\_id* is the Sequence Number Space Identifier in ASCII for the sequence number space as defined in Table 10-5 (Transmitter sequence number spaces); e.g., “SNS2” for individually addressed QoS Data

GTn is the reference start time of the EDP Epoch (see 9.4.2.337 (Enhanced Data Privacy (EDP) element) )

*Length* is the total number of bits to derive and is equal to ( 2 × *ctr\_num* × *ctr\_size* )

*start* is a state variable identifying the bit position within EDP\_SN\_offset\_block where the value starts being copied to an EDP\_SN\_offset

*finish* is a state variable identifying the bit position within EDP\_SN\_offset\_block where the value finishes being copied to an EDP\_SN\_offset

*tx* identifies the transmitter for the sequence number space,

*ctr\_index* is an index to one of the counters in the sequence number space

*ctr\_num* is the number of counters in the sequence number space

EDP\_SN\_offset( *tx*, *sns\_index*, *ctr\_index* ) is the value of EDP\_SN\_offset used for frames transmitted by *tx* using the counter identified by *ctr\_index* in the sequence number space identified by *sns\_id*

*ctr\_size* is the number of bits in the counters of the sequence number space Table 10-x provides the values of *ctr\_num* and *ctr\_size* for the sequence number spaces defined in Table 10-5 (Transmitter sequence number spaces)

Table 10-x Sequence Number values for *ctr\_num* and *ctr\_size*

|  |  |  |  |
| --- | --- | --- | --- |
| **Sequence Number Space Identifier** | **Sequence Number Space** | *ctr\_num* | *ctr\_size*  (in bits) |
| SNS1 | Baseline | TBD if an offset is needed for SNS1 | |
| SNS2 | Individually addressed QoS Data | 16 | 12 |
| SNS3 | Time Priority Management | 16 | 12 |
| SNS4 | QMF | 4 | 10 |
| SNS5 | QoS (+)Null | Not applicable. SNS5 does not have a counter | |
| SNS6 | Individually addressed PV1 Data frame | 8 | 12 |
| SNS7 | Individually addressed PV1 Management frame | 1 | 12 |
| SNS8 | Protected Fine Timing frame and Public Action LMR | TBD if an offset is needed for SNS8 | |

#### Generating EDP\_PN\_offset

For a given EDP Epoch, the set of values of EDP\_PN\_offset transmitted by the non-AP MLD and AP MLD are generated according to the following algorithm:

EDP\_PN\_offset\_block ← KDF-*Hash-Length*( KDK, “EDP\_PN\_offset”, GTn)

*start* ← 0

*finish* ← ( *PN\_size* − 1 )

**do** *tx* in {“non-AP MLD”, “AP MLD”}

EDP\_PN\_offset( *tx* ) ← EDP\_PN\_offset\_block[ *finish* : *start* ]

*start* ← ( *start* + *PN\_size* )

*finish* ← ( *finish* + *PN\_size* )

**od**

where:

EDP\_PN\_offset\_block is the KDF output subsequently partitioned into the values of EDP\_PN\_offset for both the non-AP MLD and AP MLD

KDF-*Hash-Length* is the key derivation function as defined in 12.7.1.6.2 (Key derivation function (KDF)) using the hash algorithm identified by the AKM suite selector (see Table 9-190 (AKM suite selectors))

KDK is the Key Derivation Key

GTn is the reference start time of the EDP Epoch (see 9.4.2.337 (Enhanced Data Privacy (EDP) element) )

*Length* is the total number of bits to derive and is equal to ( 2 × *PN\_size* ) = 96

*start* is a state variable identifying the bit position within EDP\_PN\_offset\_block where the value starts being copied to an EDP\_PN\_offset

*finish* is a state variable identifying the bit position within EDP\_PN\_offset\_block where the value finishes being copied to an EDP\_PN\_offset

*tx* identifies the transmitter for the sequence number space

EDP\_PN\_offset( *tx* ) is the value of EDP\_PN\_offset used for frames transmitted by *tx*

*PN\_size* is the number of bits in a Packet Number and is equal to 48