IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Proposed Text for Identifiable Random MAC, IRM | | | | |
| Date: 2021-12 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Graham SMITH | SR Technology | Sunrise, FL, USA. | 916 799 9563 | gsmith@srtrl.com |

Instructions:

Abstract

Proposed text for the Identifiable Random MAC scheme as presented in 21/1585

Rev 1 Added “No IRMK found” reason code to New IRMK request to cover case where AP has deleted “old” IRMKs. AP might delete IRMKs for time or capacity.

Added “Private” to IRM element such that an IRM STA can still use private MAC address when indicating support for IRM. This eliminates the I/G bit set to 1.

Rev 2 Correction to Element format figure.

Edits to frame names

Added IRMK Check field.

Rev 3 Use Change to prevent any brute-force attack

Typos and minor edits

Rev 4 Removed “Provide IRMK” (No good use case)

Rev 5 Edits

Rev 6 Added Clause 6

Edits to clause 11 text

Rev 7 Edits

Rev 8 Edits. Changed IRMK Check request to “may” and added Note.

Rev 9 Added IRM to Associate and Reassociate Clause 6.

Rev 10 Edits, corrections, notes

*Add following definitions to 3.2.*

**identifiable random medium access control (MAC) (IRM)**: a scheme where a non-AP STA uses identifiable random medium access control (MAC) addresses (IRMA) to prevent third parties from tracking the non-AP STA while still allowing trusted parties to identify the non-AP STA.

**identifiable random medium access control (MAC) address (IRMA):** a randomized medium access control (MAC) address used by a non-AP STA using identifiable random medium access control (MAC) (IRM).

**identifiable random medium access control (MAC) key (IRMK):** a (128-bit) key used to resolve an identifiable random medium access control (MAC) address (IRMA)

**6.3.7.2 MLME-ASSOCIATE.request**

***Add to list of primitive parameters in 6.3.7.2.2 Semantics of the service primitive***

IRM

*And add to Table*

|  |  |  |  |
| --- | --- | --- | --- |
| IRM | IRM element | As defined in (see 9.4.2.xxx IRM element) | Specifies the parameters in the IRM element. |

**6.3.7.3 MLME-ASSOCIATE.confirm**

***Add to list of primitive parameters in 6.3.7.3.2 Semantics of the service primitive***

IRM

*And add to Table*

|  |  |  |  |
| --- | --- | --- | --- |
| IRM | IRM element | As defined in (see 9.4.2.xxx IRM element) | Specifies the parameters in the IRM element. |

**6.3.7.4 MLME-ASSOCIATE.indication**

***Add to list of primitive parameters in 6.3.7.4.2 Semantics of the service primitive***

IRM

*And add to Table*

|  |  |  |  |
| --- | --- | --- | --- |
| IRM | IRM element | As defined in (see 9.4.2.xxx IRM element) | Specifies the parameters in the IRM element. |

**6.3.7.5 MLME-ASSOCIATE.response**

***Add to list of primitive parameters in 6.3.7.5.2 Semantics of the service primitive***

IRM

*And add to Table*

|  |  |  |  |
| --- | --- | --- | --- |
| IRM | IRM element | As defined in 9.4.2.xxx IRM element | Specifies the parameters in the IRM element. |

**6.3.8.2 MLME-REASSOCIATE.request**

***Add to list of primitive parameters in 6.3.8.2.2 Semantics of the service primitive***

IRM

*And add to Table*

|  |  |  |  |
| --- | --- | --- | --- |
| IRM | IRM element | As defined in 9.4.2.xxx IRM element | Specifies the parameters in the IRM element. |

**6.3.8.3 MLME-REASSOCIATE.confirm**

***Add to list of primitive parameters in 6.3.8.3.2 Semantics of the service primitive***

IRM

*And add to Table*

|  |  |  |  |
| --- | --- | --- | --- |
| IRM | IRM element | As defined 9.4.2.xxx IRM element | Specifies the parameters in the IRM element. |

**6.3.8.4 MLME-REASSOCIATE.indication**

***Add to list of primitive parameters in 6.3.8.4.2 Semantics of the service primitive***

IRM

*And add to Table*

|  |  |  |  |
| --- | --- | --- | --- |
| IRM | IRM element | As defined in 9.4.2.xxx IRM element | Specifies the parameters in the IRM element. |

**6.3.8.5 MLME-REASSOCIATE.response**

***Add to list of primitive parameters in 6.3.8.5.2 Semantics of the service primitive***

IRM

*And add to Table*

|  |  |  |  |
| --- | --- | --- | --- |
| IRM | IRM element | As defined in 9.4.2.xxx IRM element | Specifies the parameters in the IRM element. |

*Insert at end of Clause 6*

**6.3.X Identifiable random MAC (IRM)**

**6.3.X.1 General**

The Identifiable Random MAC primitives support the IRM processes as decribed in 11.xx.

**6.3.X.2 MLME-IRMKREQUEST.request**

**6.3.X.2.1 Function**

This primitive is used by an AP to transmit an IRMK Request Action frame to a specified STA.

**6.3.X.2.2 Semantics of the service primitive**

The primitive parameters are as follows:

MLME-IRMK.request(

PeerMACAddress

)

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Valid Range | Description |
| PeerMACAddress | MAC address | Any valid individual MAC address | The address of the peer MAC entity to which the IRMK Request Action frame is sent. |

**6.3.X.2.3 When generated**

This primitive is generated by the SME at an AP to request the transmission of an IRMK Request Action frame to the STA indicated by the PeerMAC Address parameter.

**6.3.X.2.4 Effect of receipt**

On receipt of this primitive, the MLME constructs an IRMK Request Action frame and then attempts to transmit this frame to the STA indicated by the PeerMACAddress parameter.

**6.3.X.3 MLME-IRMKREQUEST.confirm**

**6.3.X.3.1 Function**

This primitive reports the result of a request to send an IRMK Request Action frame.

**6.3.X.3.2 Semantics of the service primitive**

The primitive parameters are as follows:

MLME-IRMKREQUEST.confirm(

PeerMACAddress

)

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Valid Range | Description |
| PeerMACAddress | MAC address | Any valid individual MAC address | The address of the peer MAC entity to which the IRMK Request Action frame is sent. |

**6.3.X.3.3 When generated**

This primitive is generated by the MLME as a result of an MLME-IRMKREQUEST.request primitive indicating the results of that request.

**6.3.X.3.4 Effect of receipt**

The SME is notified of the results of the IRMKREQUEST.request procedure. The SME should operate according to the procedures defined in 11.xx.

**6.3.X.4 MLME-IRMKREQUEST.indication**

**6.3.X.4.1 Function**

This primitive indicates that an IRMK Request Action frame has been received.

**6.3.X.4.2 Semantics of the service primitive**

The primitive parameters are as follows:

MLME-IRMKREQUEST.indication(

PeerMACAddress

)

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Valid Range | Description |
| PeerMACAddress | MAC address | Any valid individual MAC address | The address of the peer MAC entity to which the IRMK Request Action frame is sent. |

**6.3.X.4.3 When generated**

This primitive is generated by the MLME when an IRMK Request Action frame has been received.

**6.3.X.4.4 Effect of receipt**

On receipt of this primitive, the SME either rejects the request or commences the transaction as described in 11.xx.

**6.3.X.5 MLME-IRMK.response**

**6.3.X.5.1 Function**

This primitive is used by a STA to transmit an IRMK Response Action frame to a specified AP. The IRMK Response Action frame may be transmitted as response to an IRMK Request Action frame or a New IRMK Request Action frame, or may be unsolicited.

**6.3.X.5.2 Semantics of the service primitive**

The primitive parameters are as follows:

MLME-IRMK.response(

PeerMACAddress

IRMK

)

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Valid Range | Description |
| PeerMACAddress | MAC address | Any valid individual MAC address | The address of the peer MAC entity to which the IRMK Response Action frame is sent. |
| IRMK | Integer | 0-2127 | The IRMK is a private key used to identify the STA |

**6.3.X.5.3 When generated**

This primitive is generated by the SME at a STA either in response to an IRMK RequestAction frame received from the AP indicated by the PeerMAC Address, or may be sent unsolicited.

**6.3.X.5.4 Effect of receipt**

On receipt of this primitive, the MLME constructs an IRMK Response Action frame and then attempts to transmit this frame to the AP indicated by the PeerMACAddress parameter.

**6.3.X.6 MLME-IRMKCONFIRM.request**

**6.3.X.6.1 Function**

This primitive is used by an AP to transmit an IRMK Confirm Action frame to a specified STA. The IRMK Confirm Action frame is transmitted from an AP to a non-AP STA to confirm that an IRMK has been identified.

**6.3.X.6.2 Semantics of the service primitive**

The primitive parameters are as follows:

MLME-IRMKCONFIRM.request (

PeerMACAddress

IRMK Offset

Check

)

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Valid Range | Description |
| PeerMACAddress | MAC address | Any valid individual MAC address | The address of the peer MAC entity to which the IRMK Request Action frame is sent. |
| IRMK Offset | Integer | 0-112 | A value N between 0 and 112. |
| Check | Integer | 0-254 | For n = 0 to 7 the 8 bits in Check field are:  bn = EX-OR (bN+n, bN+n+8)  where bN is Nth bit in IRMK |

**6.3.X.6.3 When generated**

This primitive is generated by the SME at an AP to request the transmission of an IRMK Confirm Action frame to the STA indicated by the PeerMAC Address parameter.

**6.3.X.6.4 Effect of receipt**

On receipt of this primitive, the MLME constructs an IRMK Confirm Action frame and then attempts to transmit this frame to the STA indicated by the PeerMACAddress parameter.

**6.3.X.7 MLME-IRMKCONFIRM.confirm**

**6.3.X.7.1 Function**

This primitive reports the result of a request to send an IRMK Confirm Action frame to a non-AP STA to confirm that an IRMK has been identified.

**6.3.X.7.2 Semantics of the service primitive**

The primitive parameters are as follows:

MLME-IRMKCONFIRM.confirm (

PeerMACAddress

IRMK Offset

Check

)

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Valid Range | Description |
| PeerMACAddress | MAC address | Any valid individual MAC address | The address of the peer MAC entity to which the IRMK Request Action frame is sent. |
| IRMK Offset | Integer | 0-112 | A value N between 0 and 112. |
| Check | Integer  (see 9.2.4.xxx.1) | 0-254 | For n = 0 to 7 the 8 bits in Check field are:  bn = EX-OR (bN+n, bN+n+8)  where bN is Nth bit in IRMK |

**6.3.X.7.3 When generated**

This primitive is generated by the MLME as a result of an MLME-IRMKCONFIRM.request primitive indicating the results of that request.

**6.3.X.7.4 Effect of receipt**

The SME is notified of the results of the IRMKCONFIRM.request procedure. The SME should operate according to the procedures defined in 11.xx

**6.3.X.8 MLME-NEWIRMK.request**

**6.3.X.8.1 Function**

This primitive is used by an AP to transmit a New IRMK Request Action frame to a specified STA. The New IRMK Request Action frame is transmitted from an AP to a non-AP STA to request that the STA submits an IRMK. The expected response to a New IRMK Request Action frame is an IRMK Response Action frame.

**6.3.X.8.2 Semantics of the service primitive**

The primitive parameters are as follows:

MLME-NEWIRMK.request (

PeerMACAddress

IRMK Reason

)

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Valid Range | Description |
| PeerMACAddress | MAC address | Any valid individual MAC address | The address of the peer MAC entity to which the New IRMK Request Action frame is sent. |
| IRMK Reason | Integer | 0-254 | Specifies the reason that the AP is sending the New IRMK request as shown in Table 9ddd. |

**6.3.X.8.3 When generated**

This primitive is generated by the SME at an AP to request the transmission of a New IRMK Request Action frame to the STA indicated by the PeerMAC Address parameter.

**6.3.X.8.4 Effect of receipt**

On receipt of this primitive, the MLME constructs a New IRMK Request Action frame according to procedure in 11.xxx, and then attempts to transmit this frame to the STA indicated by the PeerMACAddress parameter.

**6.3.X.9 MLME-NEWIRMK.confirm**

**6.3.X.9.1 Function**

This primitive reports the result of a request to send a New IRMK Request Action frame.

**6.3.X.9.2 Semantics of the service primitive**

The primitive parameters are as follows:

MLME-NEWIRMK.confirm (

PeerMACAddress

IRMK Reason

)

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Valid Range | Description |
| PeerMACAddress | MAC address | Any valid individual MAC address | The address of the peer MAC entity to which the New IRMK Request Action frame is sent. |
| IRMK Reason | Integer | 0-254 | Specifies the reason that the AP is sending the New IRMK request as shown in Table 9ddd. |

**6.3.X.9.3 When generated**

This primitive is generated by the MLME as a result of an MLME-NEWIRMK.request primitive indicating the results of that request.

**6.3.X.9.4 Effect of receipt**

The SME is notified of the results of the NEWIRMK.request procedure. The SME should operate according to the procedures defined in 11.xx.

**6.3.X.10 MLME-NEWIRMK.indication**

**6.3.X.10.1 Function**

This primitive indicates that a New IRMK Request Action frame has been received.

**6.3.X.10.2 Semantics of the service primitive**

The primitive parameters are as follows:

MLME-NEWIRMK.indication (

PeerMACAddress

IRMK Reason

)

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Valid Range | Description |
| PeerMACAddress | MAC address | Any valid individual MAC address | The address of the peer MAC entity to which the New IRMK Request Action frame is sent. |
| IRMK Reason | Integer | 0-254 | Specifies the reason that the AP is sending the New IRMK request as shown in Table 9ddd. |

**6.3.X.10.3 When generated**

This primitive is generated by the MLME when a New IRMK Request Action frame has been received.

**6.3.X.10.4 Effect of receipt**

On receipt of this primitive, the SME either rejects the request or commences the transaction as described in 11.xx.

**6.3.X.11 MLME-IRMKCHECK.request**

**6.3.X.11.1 Function**

This primitive is used by an AP to transmit an IRMK Check Request Action frame to a specified STA. The IRMK Check Request Action frame is transmitted from an AP to a non-AP STA to request that the STA submits an IRMK Check Response frame.

**6.3.X.11.2 Semantics of the service primitive**

The primitive parameters are as follows:

MLME-IRMKCHECK.request (

PeerMACAddress

)

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Valid Range | Description |
| PeerMACAddress | MAC address | Any valid individual MAC address | The address of the peer MAC entity to which the IRMK Check Request Action frame is sent. |

**6.3.X.11.3 When generated**

This primitive is generated by the SME at an AP to request the transmission of an IRMK Check Request Action frame to the STA indicated by the PeerMAC Address parameter.

**6.3.X.11.4 Effect of receipt**

On receipt of this primitive, the MLME constructs an IRMK Check Request Action frame according to procedure in 11.xxx, and then attempts to transmit this frame to the STA indicated by the PeerMACAddress parameter.

**6.3.X.12 MLME-IRMKCHECK.confirm**

**6.3.X.12.1 Function**

This primitive reports the result of a request to send an IRMK Check Request Action frame.

**6.3.X.12.2 Semantics of the service primitive**

The primitive parameters are as follows:

MLME-IRMKCHECK.confirm (

PeerMACAddress

)

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Valid Range | Description |
| PeerMACAddress | MAC address | Any valid individual MAC address | The address of the peer MAC entity to which the IRMK Check Request Action frame is sent. |

**6.3.X.12.3 When generated**

This primitive is generated by the MLME as a result of an MLME-IRMKCHECK.request primitive indicating the results of that request.

**6.3.X.12.4 Effect of receipt**

The SME is notified of the results of the IRMKCHECK.request procedure. The SME should operate according to the procedures defined in 11.xx.

**6.3.X.13 MLME-IRMKCHECK.indication**

**6.3.X.13.1 Function**

This primitive indicates that an IRMK Check Request Action frame has been received.

**6.3.X.13.2 Semantics of the service primitive**

The primitive parameters are as follows:

MLME-IRMKCHECK.indication (

PeerMACAddress

)

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Valid Range | Description |
| PeerMACAddress | MAC address | Any valid individual MAC address | The address of the peer MAC entity to which the IRMK Check Request Action frame is sent. |

**6.3.X.13.3 When generated**

This primitive is generated by the MLME when an IRMK Check Request Action frame has been received.

**6.3.X.13.4 Effect of receipt**

On receipt of this primitive, the SME either rejects the request or commences the transaction as described in 11.xx.

**6.3.X.14 MLME-IRMKCHECK.response**

**6.3.X.14.1 Function**

This primitive is used by a STA to transmit an IRMK Check Response Action frame to a specified AP in response to an IRMK Check Request Action frame

**6.3.X.14.2 Semantics of the service primitive**

The primitive parameters are as follows:

MLME-IRMKCHECK.response (

PeerMACAddress

IRMK Offset

Check

)

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Valid Range | Description |
| PeerMACAddress | MAC address | Any valid individual MAC address | The address of the peer MAC entity to which the IRMK CHECK Response Action frame is sent. |
| IRMK Offset | Integer | 0-112 | A value N between 0 and 112. |
| Check | Integer  (see 9.2.4.xxx.1) | 0-254 | For n = 0 to 7 the 8 bits in Check field are:  bn = EX-OR (bN+n, bN+n+8)  where bN is Nth bit in IRMK |

**6.3.X.14.3 When generated**

This primitive is generated by the SME at a STA in response to an IRMK Check Request Action frame from the AP indicated by the PeerMAC Address.

**6.3.X.14.4 Effect of receipt**

On receipt of this primitive, the MLME constructs an IRMK Response Action frame and then attempts to transmit this frame to the AP indicated by the PeerMACAddress parameter.

*Insert new row in Table 9-79 Action field Clause 9.4.1.11*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Code** | **Meaning** | **See subclause** | **Robust** | **Group addressed Privacy** |
| <ANA> | IRM | 9.6.aa | Yes | No |
| <ANA> -125 | Reserved |  |  |  |

*Insert new row in Table 9-190 Extended Capabilities field, Clause 9.4.2.26*

|  |  |  |
| --- | --- | --- |
| **Bit** | **Information** | **Notes** |
| <ANA> | IRM Capability | The STA sets IRM Capability subfield to 1 to indicate support for IRM and sets to 0 if IRM is not supported. |

*Insert new row in Table 9-128 – Element IDs*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Element ID** | **Element ID Extension** | **Extensible** | **Fragmentable** |
| IRM (see 9.4.2.xxx IRM element) | 255 | <ANA> | No | No |

*Insert new row in Table 9-62 – Association Request frame body*

|  |  |  |
| --- | --- | --- |
| **Order** | **Information** | **Notes** |
| <ANA> | IRM | The IRM element is present if IRM Capability subfield is set to 1. |

*Insert new row in Table 9-64 – Reassociation Request frame body*

|  |  |  |
| --- | --- | --- |
| **Order** | **Information** | **Notes** |
| <ANA> | IRM | The IRM element is present if IRM Capability subfield is set to 1. |

*Insert new clause 9.4.2.xxx*

**9.4.2.xxx Identifiable Random MAC (IRM) element**

The IRM element is used by a non-AP STA that is using an IRMA. The format of the IRM element is defined in Figure 9–yyy.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Element ID | Length | Element ID Extension | IRM Indicator | IRM Hash  (Not present if IRM Indicator set to 0) | IRMK Check  (Optional) |

Octets: 1 1 1 1 (16) (2)

**Figure – 9-yyy – IRM element format**

The Element ID, Element ID Extension and Length fields are defined in 9.4.2.1 (General)

The IRM Indicator field indicates IRM related information as defined in Table 9 – zzz

**Table 9–zzz – IRM Indicator**

|  |  |  |
| --- | --- | --- |
| **IRM Indicator field value** | **Field name** | **Notes** |
| 0 | Private | A non-AP STA sets the IRM Indicator field value to 0 to indicate that the non-AP STA is using a private random MAC address, i.e., is not using an IRMA |
| 1 | Unknown | A non-AP STA sets the IRM Indicator field value to 1 to indicate that the non-AP STA has not previously provided an IRMK to the AP |
| 2 | Known | A non-AP STA sets the IRM Indicator field value to 2 to indicate that the non-AP STA has previously provided an IRMK to the AP |
| 3 | Change | A non-AP STA sets the the IRM Indicator field value to 3 to indicate that the non-AP STA has previously provided an IRMK to the AP but will change the IRMK once associated |
| 3-255 | Reserved |  |

The IRM Hash field is not present if the IRM Indicator field is set to “Private”. The IRM Hash field is a (128-bit) hash that is derived from the IRMA and the IRMK as defined in 11.xx.2.

The IRM Check field is defined in 9.2.4.xxx.1.

**9.2.4.xxx.1 IRM Check field**

The IRM Check field is optionally present in the IRM element if the IRM Indicator field is set to Known or Change and is preset in the IRM Confirm Action field.

The format of the IRM Check field is shown in Figure 9-jjj

|  |  |
| --- | --- |
| IRMK Offset | Check |

Octets: 1 1

**Figure – 9-jjj – IRMK Check field format**

The IRMK Offset field has a value N between 0 and 112.

The Check field contains 8 bits representing the EX-OR of the 8 bits of the IRMK, bN to bN+7 with the following 8 bits (bN+8 to bN+15).

i.e. For n = 0 to 7 the 8 bits in Check field are:

bn = EX-OR (bN+n, bN+n+8) where bN is Nth bit in IRMK

Note: As an example, if the IRMK Offset field has a value of 72, then the Check field b0 is EX\_OR of b72 and b80 of the IRMK, and Check field b7 is EX-OR of b79 and b87 of the IRMK.

*Insert new row to Table 9-404 - ANQP-element definitions*

|  |  |  |
| --- | --- | --- |
| **ANQP-element name** | **InfoID** | **ANQP-element (subclause)** |
| Identifiable Random MAC (IRM) | <ANA> | 9.4.5.aaa |

*Insert new Clause 9.4.5.aaa*

**9.4.5.aaa Identifiable Random MAC (IRM) ANQP-element**

The IRM ANQP-element is used by a non-AP STA that is using an IRMA and has previously provided an IRMK to that AP. The format of the IRM ANQP-element is defined in Figure 9 – xyz.

|  |  |  |  |
| --- | --- | --- | --- |
| Info ID | Length | IRM Hash | IRM Check  (optional) |

Octets: 1 1 16 2

**Figure – 9-xyz – IRM ANQP-element format**

The Info ID and Length fields are defined in 9.4.5.1 (General)

The IRM Hash field is a (128-bit) hash that is derived from the IRMA and the IRMK as defined in 11.xx.2.

The IRM Check field is optionally present and is defined in 9.4.xx.1.

*Insert new clause at end of 9.6 Action frame format details*

**9.6.aa IRM Action frame details**

**9.6.aa.1 General**

Several Action frame formats are defined for IRM purposes. These frames are identified by the single octet IRM Action field, which follows immediately after the Category field. The values of the IRM Action field are defined in Table 9-bbb (IRM Action field).

**Table 9-bbb – IRM Action field**

|  |  |
| --- | --- |
| Action field value | Meaning |
| 0 | IRMK Request |
| 1 | IRMK Response |
| 2 | IRMK Confirm |
| 3 | New IRMK Request |
| 4 | IRMK Check Request |
| 5 | IRMK Check Response |
| 6-255 | Reserved |

**9.6.aa.2 IRMK Request**

The IRMK Request Action frame is transmitted by an AP to a non-AP STA that associated to the AP with the IRM Capability bit set to 1 in the Extended Capabilities field, the IRM element includes an IRM Hash field, and the IRM Indicator field in the IRM element is set to “Unknown”. The format of the IRMK Request Action field is shown in Figure 9-ccc.

|  |  |
| --- | --- |
| Category | IRM Action |

Octets: 1 1

**Figure – 9-ccc – IRMK Request Action field format**

The Category field is defined in 9.4.1.1.1(Action field)

The IRM Action field is defined in Table 9-bbb in 9.6.aa.1 (General).

**9.6.aa.3 IRMK Response**

The IRMK Response Action frame is transmitted from a non-AP STA to an AP in response to an IRMK Request frame. An unsolicited IRMK Response Action frame may be transmitted from a non-AP STA to an AP. The format of the IRMK Request Action field is shown in Figure 9-ddd.

|  |  |  |
| --- | --- | --- |
| Category | IRM Action | IRMK |

Octets: 1 1 16

**Figure – 9-ddd – IRMK Response Action field format**

The Category field is defined in 9.4.1.1.1(Action field)

The IRM Action field is defined in Table 9-bbb in 9.6.aa.1 (General).

The IRMK field is a (128-bit) key that is used together with the IRMA to derive the value of the IRM Hash field that is sent in the IRM element.

**9.6.aa.4 IRMK Confirm**

The IRMK Confirm Action frame is transmitted from an AP to a non-AP STA to confirm that an IRMK has been identified. The format of the IRMK Confirm Action field is shown in Figure 9-eee.

|  |  |  |
| --- | --- | --- |
| Category | IRM Action | IRMK Check |

Octets: 1 1 2

**Figure – 9-eee – IRMK Confirm Action field format**

The Category field is defined in 9.4.1.1.1(Action field)

The IRM Action field is defined in Table 9-bbb in 9.6.aa.1 (General).

The IRM Check field is defined in 9.2.4.xxx.1.

**9.6.aa.5 New IRMK Request**

The New IRMK Request Action frame is transmitted by an AP to a non-AP STA when an AP requests the non-AP STA to provide a new IRMK. The format of the New IRMK Request Action field is shown in Figure 9-iii.

|  |  |  |
| --- | --- | --- |
| Category | IRM Action | IRMK Reason |

Octets: 1 1 1

**Figure – 9-iii – New IRMK Request Action field format**

The Category field is defined in 9.4.1.1.1(Action field)

The IRM Action field is defined in Table 9-bbb in 9.6.aa.1 (General).

The values of the IRMK Reason field are defined in Table 9-ddd

**Table 9-ddd – IRMK Reason field values**

|  |  |
| --- | --- |
| **IRMK Reason field value** | **Meaning** |
| 0 | No reason provided |
| 1 | Non-AP STA requested change |
| 2 | No IRMK found |
| 3 | Duplicate Key exists |
| 4 | Key not random |
| 5-255 | Reserved |

**9.6.aa.6 IRMK Check Request**

The IRMK Check Request Action frame is transmitted by an AP to a non-AP STA when an AP requests the non-AP STA to provide an IRMK Check. The format of the IRMK Request Action field is shown in Figure 9-jjj.

|  |  |
| --- | --- |
| Category | IRM Action |

Octets: 1 1

**Figure – 9-jjj–IRMK Check Request Action field format**

The Category field is defined in 9.4.1.1.1(Action field)

The IRM Action field is defined in Table 9-bbb in 9.6.aa.1 (General).

**9.6.aa.7 IRMK Check Response**

The IRMK Check Response Action frame is transmitted from a non-AP STA to an AP in response to an IRMK Check Request Action frame. The format of the IRMK Check Response Action field is shown in Figure 9-kkk.

|  |  |  |
| --- | --- | --- |
| Category | IRM Action | IRMK Check |

Octets: 1 1 2

**Figure – 9-kkk – IRMK Confirm Action field format**

The Category field is defined in 9.4.1.1.1(Action field)

The IRM Action field is defined in Table 9-bbb in 9.6.aa.1 (General).

The IRM Check field is defined in 9.2.4.xxx.1.

*Add a new subclause at the end of clause 11 (MLME)*

**11.xx Identifiable random MAC (IRM) operation**

**11.xx.1 General**

To mitigate tracking and traffic analysis, a non-AP STA may randomly change its MAC address (see 4.5.4.10). For some services, however, it may be desirable to the user that the non-AP STA is identified by the AP and network services. IRM operation enables a non-AP STA to use an identifiable random MAC address for every (re)association. An AP can then store a list of identified non-AP STAs and a non-AP STA can store a list of identities and APs.

A non-AP STA advertises support for IRM by setting the IRM Capability subfield to 1 in the Extended Capabilites element in Probe Request, Association Request and Reassociation Request frames. A non-AP STA includes an IRM element in its Association Request and Reassociation Request frames. An AP advertises support for IRM by setting the IRM Capability subfield to 1 in the Extended Capabilites element in its Beacon and Probe Response frames.

To indicate that the non-AP STA intends to be identifiable, the non-AP STA includes, in the IRM element, the IRM Hash field and sets the IRM Indicator field to “Unknown”, “Known”, or “Change”. If the non-AP STA includes an IRM Hash field, the randomized MAC address is an IRM Address (IRMA). If the non-AP STA intends that it not be identifiable, the IRM element does not include an IRM Hash field and the IRM Indicator field is set to “Private”.

If the non-AP STA intends that it be identifiable, and the IRM element includes an IRM Hash field, the non-AP STA generates a 128-bit IRM key (IRMK) which may be constant or may vary for each SSID or AP or ESS. In an Association Request or Reassociation Request, the non-AP STA uses an IRMA as the TA, and includes the IRM element with an IRM Hash field and optionally an IRMK Check field. The IRM Hash field value is derived from the IRMA and the IRMK (see 11.xx.2).

When a non-AP STA associates with an IRMA as the TA, and indicates “Unknown”, once the non-AP STA is associated, the AP shall send an IRMK Request Action frame and the non-AP STA shall respond with an IRMK Response Action frame that contains the IRMK. The AP should store that IRMK as an identifier for that non-AP STA. When a non-AP STA (re)associates with an IRMA as the TA, and indicates “Known” or “Change”, the AP shall calculate the IRM Hash value using each of its stored IRMKs until it finds the IRMK that produces the same IRM Hash value as that included in the IRM element. If the non-AP STA included an IRMK Check field in the IRM element, then the AP can reduce the number of IRMKs that need to be checked. The IRMK acts as the identifier for the non-AP STA. Once associated, a non-AP STA may change its IRMK.

A list of IRMKs and non-AP STAs may be stored by the AP and used as an identifier for each non-AP STA. A non-AP STA may store the IRMK exchanged with a particular AP such that each time the non-AP STA associates to that AP, the AP can identify the non-AP STA. If the non-AP STA changes the IRMK, then the AP shall update its list, such that the particular non-AP STA is still identifiable.

**11.xx.2 Identifiable random MAC (IRM) Address**

A non-AP STA that supports IRM and that intends to be identified, (re)associates to an AP that also supports IRM, using an identifiable random MAC address (IRMA) as its TA. An IRMA is a randomized MAC address constructed from the locally administered address space (see 12.2.10). To indicate that the non-AP STA intends to be identifiable, the IRM Hash field is included in the IRM element.

**11.xx.2 Identifiable random MAC (IRM) Hash**

The IRM Hash field value is the SHA-256/128 function of the IRMK and the IRMA.

IRM Hash = SHA-256/128 (IRMK, IRMA)

The 128-bit IRM Hash field is included in the IRM element and the IRM-ANQP element when the non-STA intends to be identifiable.

Note: SHA-256/128 is the truncated SHA-256 where the leftmost 128 bits of the 256-bit hash generated by SHA-256 are selected as the truncated 128 bit IRM Hash.

**11.xx.3 Identifiable random MAC (IRM) association**

A non-AP STA that supports IRM shall include the IRM element in Association and Reassociation Request frames.

If the non-AP STA has not previously provided an IRMK to the AP, then the IRM Indicator field value in the IRM element in the Association Request frame, shall be set to “Unknown”. If the non-AP STA has previously provided an IRMK to the AP, and intends to change the IRMK immediately after association the IRM Indicator field shall be set to “Change”, otherwise the IRM Indicator field shall be set to “Known”. If the IRM Indicator is set to “Known” or “Change”, the IRMK Check field may be included in the IRM element.

If the non-AP STA is not associating with an IRMA but with a private randomized MAC address, then the IRM Indicator field value in the IRM element in the Association or Reassociation Request frame, shall be set to “Private” and neither the IRM Hash field nor the IRMK Check field shall be present in the IRM element.

If a non-AP STA associates to an AP using an IRMA and the IRM Indicator field is set to “Unknown”, the non-AP STA constructs an IRMK and calculates the IRM Hash value (see 11.xx.2) which is included in the IRM element in the Association Request. Once associated, the AP shall transmit an IRMK Request Action frame to the non-AP STA and the non-AP STA shall send an IRMK Response Action frame that includes the IRMK, to the AP. The AP may optionally, as a check, calculate the IRM Hash value using the IRMA and the IRMK provided by the non-AP STA, and confirm that it is identical to the IRM Hash provided by the non-AP STA in the IRM element. The AP shall then store the IRMK as the identifier for that non-AP STA.

If, in an (Re)Association Request frame, the IRM Indicator field value in the IRM element is set to “Known”, or “Change”, the non-AP STA shall also include in the IRM element the IRM Hash field and may include the IRMK Check field. The IRMK Check field may be used by an AP to down-select stored IRMKs and reduce the number of hash calculations required in order to find the correct IRMK. The AP may, prior to association, check through its stored IRMKs in order to determine the IRMK that, together with the IRMA, produces the IRM Hash value that the non-AP STA included in the IRM element in the (Re)Association Request frame. Alternatively, the AP shall, after association, check its stored IRMKs in order to determine the IRMK that, together with the IRMA, produces the IRM Hash value that the non-AP STA included in the IRM element. After the non-AP STA has associated, and the AP has determined the IRMK, the AP may transmit an IRMK Check Request Action frame to the non-AP STA and the non-AP STA shall respond with an IRMK Check Response Action frame. In this case, if the non-AP STA did include an IRMK Check field in the IRM element, then the IRMK Offset field in the IRMK Check Response Action frame shall be a different value to the IRMK Offset field in the IRM element. The AP shall then confirm that the IRMK it has stored for that STA is correct. If it is not correct then the AP shall disassociate the STA.

Note: The AP might use the IRMK Check Request Action field as a challenge to the non-AP STA to counter a third party non-AP STA that copies the IRMA and IRM Hash and also has access to the network, i.e., the network security is weak, for example where the password is on display. In secure networks this challenge might not be required.

If, the non-AP STA did not include an IRMK Check field in the IRM element, then, after association, the AP may request an IRMK Check field by sending an IRMK Check Request Action frame to the non-AP STA. The non-AP STA shall then respond with an IRMK Response Action frame which includes the IRMK Check field.   
Note: An AP that has many stored IRMKs, might request an IRMK Check field if the non-AP STA omitted it from the IRM element in the (re)assocation request. The IRMK Check can reduce the number of IRMKs that the AP need check, by a factor of 256.

If, in the Association or Reassociation Request frame, the IRM Indicator field value is set to “Change”, once the non-AP STA is associated and the AP has determined and confirmed the IRMK for the non-AP STA, the AP shall transmit a New IRMK Request Action frame to the non-AP STA, with the IRMK Reason field set to “non-AP STA requested change”. The non-AP STA shall then transmit an IRMK Response frame with the new IRMK. The AP shall then use this IRMK as the new identifier for the non-AP STA.

When associated, a non-AP STA may send an unsolicited IRMK Response Action frame to the AP in order to change its IRMK.

The IRMA and IRM Hash should change for every (re)association.

Note: To identify a non-AP STA, a third party would need to go determine the IRMK, which requires, on average, 2127 or, if an IRMK Check field has been sent, 2119 hash calculations. In the unlikely event that a third party has found the IRMK, as the IRMA and IRM Hash change every association, the third party needs to confirm the IRMK every association to establish if this is the same non-AP STA that has previously been associated to this AP. If, however, once associated, the non-AP STA changes its IRMK, the next time that the non-AP STA associates, it is then impossible for a third party to establish if this is the same non-AP STA.

An AP might delete IRMKs from its stored list for various reasons e.g., time, capacity. If a non-AP STA sets the IRM Indicator field in the IRM element to “Known”, and the AP does not find a corresponding IRMK, then the AP shall request the associated non-AP STA to provide an IRMK by sending a New IRMK Request Action frame to the non-AP STA with the IRM Reason field set to “No IRMK found”. The non-AP STA may then respond with an IRMK Response Action frame that includes a new IRMK or may provide the orginal IRMK, or may choose not to provide an IRMK.

An AP may request an associated non-AP STA to provide a new IRMK by sending a New IRMK Request Action frame to the non-AP STA. The AP shall include a reason for the request in the IRMK Reason field, see Table 9-ccc, and the non-AP STA may either respond with an IRMK Response Action frame that includes a new IRMK to be used as its identifier, or may ignore the request, or may take other action, such as disassociate.

**11.xx.4 Identifiable random MAC (IRM) pre-association**

A non-AP STA, using an IRMA as the TA, that has previously exchanged an IRMK with an AP, may send an IRM ANQP-element to that AP that contains the IRM Hash field and may contain an IRM Check field, such that the AP can identify the non-AP STA pre association using the stored IRMKs.

A non-AP STA, using an IRMA as the TA, that has previously exchanged an IRMK with an AP, may include the IRM element in a directed probe to the AP with the IRM indicator field set to “Known”.

Note: By sending a directed probe a non-AP STA might advertise its presence to an AP before or without the need to associate.

**11.xx.5 Stored IRMKs**

An AP maintains a list of stored IRMKs and non-AP STAs. The AP can use this list to identify a specific non-AP STA to an IRMK. The AP may determine further information or IDs about an associated non-AP STA such as membership number, guest information, family member, subscription, etc. The gathering and determination such IDs may be out of scope.