IEEE P802.11
Wireless LANs

|  |
| --- |
| Opportunistic Wireless Encryption |
| Date: 2015-09-15 |
| Author(s): |
| Name | Affiliation | Address | Phone | email |
| Dan Harkins | Aruba Networks, an HP Company | 1322 Crossman avenue, Sunnyvale, California, United States of America | +1 408 225 4500 | dharkins at aruba networks dot com |
| Warren Kumari | Google, Inc. | 1600 Amphitheatre Parkway, Mountain View, California, United States of America | +1 650 253 0000 | warren at kumara dot net |

Abstract

This submission proposes an opportunistic encryption scheme for 802.11.

***Instruct the editor to modify section 4.10.3.1 as indiated:***

**4.10.3.1 General**

This subclause summarizes the system setup and operation of an RSN, in fourcases: when a password or PSK is used during IEEE Std 802.11 authentication, when an IEEE Std 802.1X AS is used after Open System authentication, when a PSK is used after Open System authentication, and when a Diffie-Hellman exchange is opportunistically used during IEEE Std 802.11 (re)association. For an ESS, the AP includes an Authenticator, and each associated STA includes a Supplicant.

***Instruct the editor to create a new section 4.10.3.5:***

**4.10.3.5 Opportunistic Wireless Encryption**

The following AKM operations are carried out when opportunistic encryption is accomplished using a Diffie-Hellman key exchange.

* The STA discovers the AP’s security policy indicating support for Opportunistic Wireless Encryption through passively monitoring Beacon frames or through actively probing. After discovery in a non-DMG Infrastructure BSS, the STA performs Open System Authentication with the AP.
* The STA (re)associates with the AP and provides an ephemeral Diffie-Hellman public key in its (Re)Association Request frame. The AP responds with an ephemeral Diffie-Hellman public key in its (Re)Association Response frame.
* The STA and AP complete the Diffie-Hellman key exchange and generate a PMK.
* The 4-Way Handshake using EAPOL-Key frames is used, just as with IEEE Std 802.1X authentication, when an AS is present. See Figure 4-27 (Establishing pairwise and group keys).
* The GTK and GTK sequence number are sent from the AP to the STA just as in the AS case. See Figure 4-27 (Establishing pairwise and group keys) and Figure 4-28 (Delivery of subsequent group keys).
* If management frame protection is negotiated, the IGTK and IGTK packet number are sent from the AP to the STA just as in the AS case. See Figure 4-27 (Establishing pairwise and group keys) and Figure 4-28 (Delivery of subsequent group keys).

***Instruct the editor to modify sections 8.3.3.5-8.3.3.8 and replace ANA-1 with the next available number:***

**8.3.3.5 Association Request frame format**

 **Table 8-29—Association Request frame body**

|  |  |  |
| --- | --- | --- |
|  Order |  Information |  Notes |
| <ANA-1> |  Diffie-Hellman Parameter | The Diffie-Hellman Parameter is present if dot11RSNAActivated is true and the OWE AKM is indicated in the RSNE in the Association Request frame. |

**8.3.3.6 Association Response frame format**

 **Table 8-30—Association Response frame body**

|  |  |  |
| --- | --- | --- |
|  Order |  Information |  Notes |
| <ANA-1> |  Diffie-Hellman Parameter | If the Status code is zero and the Diffie-Hellman Parameter is present if dot11RSNAActivated is true and the OWE AKM is indicated in the RSNE in the Association Request frame to which this frame is in response to. |

**8.3.3.7 Reassociation Request frame format**

 **Table 8-31—Reassociation Request frame body**

|  |  |  |
| --- | --- | --- |
|  Order |  Information |  Notes |
| <ANA-1> |  Diffie-Hellman Parameter | The Diffie-Hellman Parameter is present if dot11RSNAActivated is true and the OWE AKM is indicated in the RSNE in the Reassociation Request frame. |

**8.3.3.8 Reassociation Response frame format**

 **Table 8-32—Reassociation Response frame body**

|  |  |  |
| --- | --- | --- |
|  Order |  Information |  Notes |
| <ANA-1> |  Diffie-Hellman Parameter | If the Status code is zero and the Diffie-Hellman Parameter is present if dot11RSNAActivated is true and the OWE AKM is indicated in the RSNE in the Reassociation Request frame to which this frame is in response to. |

***Instruct the editor to insert a new row in table 8-74 and request allocation of a number from ANA replacing “<ANA-2>” with the assigned number:***

**8.4.2.1 General**

 **Table 8-74—Element IDs**

|  |  |  |  |
| --- | --- | --- | --- |
|  Element |  Element ID |  Element ID Extension |  Extensible |
|  Diffie-Hellman Parameter |  255 |  <ANA-2> |  No |

***Instruct the editor to insert a new row in table 8-130 and request allocation of a number from ANA replacing “<ANA-3>” with the assigned number:***

**8.4.2.24.3 AKM Suites**

 **Table 8-130—AKM suite selectors**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  OUI |  Suite Type | Authentication Type |   Key Management type | Key derivation type |
|  00-0F-AC | <ANA-3> | Opportunistic Wireless Encryption  | RSNA key management defined in 11.6 (Keys and key distribution) with SHA-256 | Defined in 11.6.1.7.2 (Key derivation function (KDF)) using SHA-256 |

***Instruct the editor to create a new section 8.4.2.171:***

**8.4.2.171 Diffie-Hellman Parameter element**

The Diffie-Hellman Parameter element contains a Diffie-Hellman public value and an indicator of the finite cyclic group from which it was obtained. See figure 8-XYZ (Diffie-Hellman Parameter element).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  Element ID |  Length | Element ID Extension |  Group |  Public Key |

 Octets: 1 1 1 2 variable

The Element ID and Length fields are defined in 8.4.2.1 (General).

The Group field is a 16-bit unsigned integer that maps an identifying number from the “Group Description” registry maintained by IANA for IETF RFC 2409 (IKE) to a complete domain parameter set.

The Public Key is a Diffie-Hellman public key, an element in the group described by the domain parameter set indicated by the value in the Group field.

***Instruct the editor to modify section 11.5.10.3 as indicated:***

**11.5.10.3 Cached PMKSAs and RSNA key management**

In a non-FT environment, a STA might retain PMKSAs it establishes as a result of previous authentication. The PMKSA cannot be changed while cached. The PMK in the PMKSA is used with the 4-Way Handshake to establish fresh PTKs.

If a STA in an ESS has determined it has a valid PMKSA with an AP to which it is about to (re)associate, it performs Open System authentication to the AP, and then it includes the PMKID for the PMKSA in the RSNE in the (Re)Association Request. A STA performing PMKSA cachine shall not indicate the OWE Authentication method in the RSNE in the (Re)Association Request. When the PMKSA was not created using pre-authentication, the AKM indicated in the RSNE by the STA in the (Re)Association Request shall be identical to the AKM used to establish the cached PMKSA in the first place.

***Instruct the editor to create a new section 11.6.12, and the following sub-sections:***

**11.6.12 Opportunistic Wireless Encryption Handshake**

**11.6.12.1 General**

The Opportunistic Wireless Encryption is executed between a non-AP STA and an AP to establish a PMKSA using a simple Diffie-Hellman key exchange. The exchange does not provide true authentication of the non-AP STA or AP but does allow for encryption. It is designed for cases in which access control is either not necessary or can be handled outside of this standard.

**11.6.12.2 Opportunistic Wireless Encryption exchange**

The Opportunistic Wireless Encryption (OWE) exchange occurs during association of a non-AP STA and an AP that indicate the OWE AKM (8.4.2.24.3 AKM Suites) by placing a Diffie-Hellman Parameter element in the (Re)Association Request and Response frames, respectively.

First the non-AP STA chooses a group in which to perform the Diffie-Hellman exchange. It then generates a secret private key, *m*, by randomly choosing it such that 1 < *m* < *r*, where *r* is the (prime) order of the chosen group. It derives a public key, ***M***, from the private key and the generator from the chosen group, ***G***:

 ***M*** = scalar-op(*m*, ***G***)

Where scalar-op() is defined in 11.3.4 (Finite cyclic groups). The non-AP STA then constructs a Diffie-Hellman Parameter element by assigning the appropriate value from the IANA-maintained registry for the chosen group to the Group field and assigning the public key to the Public key field using the conversion defined in 11.3.7.2.4 (Element to octet string conversion). The non-AP STA inserts the Diffie-Hellman Parameter element into its (Re)Association Request frame per 8.3.3.5 (Association Request frame format) or 8.3.3.7 (Reassociation Request frame format) and transmits the (Re)Association Request frame to the AP.

Upon receipt of the non-AP STA’s (Re)Association Request frame indicating OWE, the AP checks the Group field of the Diffie-Hellman Parameter element. If it is not an acceptable value the AP shall reject association with the reason of UNSUPPORTED\_FINITE\_CYCLIC\_GROUP. Otherwise it shall extract the non-AP STA’s public key from the Public Key field using the conversion defined in 11.3.7.2.5 (Octet string to element conversion). The extracted public key, ***M***, shall then be validated using the element validation technique from 11.3.5.4 (Processing of a peer’s SAE Commit message). If validation of the public key fails, the AP shall reject association with the reason of REQUEST\_DECLINED. If validation of the public key is successful, (Re)Association continues.

The AP shall then generate a private key, *n*, by randomly choosing it such that 1 < *n* < *r*, where *r* is the (prime) order of the group indicated in the Group field of the received Diffie-Hellman Parameter element. It derives a public key, ***N***, from the private key and the generator from the chosen group, ***G***:

 ***N*** = scalar-op(*n*, ***G***)

Where scalar-op() is defined in 11.3.4 (Finite cyclic groups). It shall then complete the Diffie-Hellman key exchange and generate a shared secret a PMK, and a PMKID:

 ***S*** = scalar-op(*n*, ***M***)

 *s* = F(***S***)

 *PMK* = KDF-Hash-256(s, “OWE PMK Generation”, F(***M***) || F(***N***))

 *PMKID* = Truncate-128(SHA-256(F(***M***) || F(***N***)))

Where KDF-Hash-256 is the key derivation function defined in 11.6.1.7.2 (Key derivation function (KDF)) using the hash algorithm defined by the AKM in Table 8-131 to generate a 256-bit key, and F() is the element-to-scalar mapping function from 11.3.4 (Finite Cyclic groups). Intermediate data, ***S*** and *s*, shall be irretrievably deleted upon generation of the PMK. Upon generation of the PMKID the public keys ***M*** and ***N*** can be deleted. A PMKSA shall be created containing the PMK and PMKID.

The AP shall then construct an (Re)Association Response frame per 8.3.3.6 (Association Response frame format) or 8.3.3.8 (Reassociation Response frame format) indicating OWE. The AP constructs a Diffie-Hellman Parameter element with the Group field being identical to the Group field of the received (Re)Association Request frame and assigning its public key to the Public key field using the conversion defined in 11.3.7.2.4 (Element to octet string conversion). The AP inserts the Diffie-Hellman Parameter element into its (Re)Association Response frame per 8.3.3.6 (Association Response frame format) or 8.3.3.8 (Reassociation Response frame format) and transmits the (Re)Association Response frame to the STA.

Upon receipt of the AP’s (Re)Association Response frame indicating OWE, the STA checks that the Group field of the Diffie-Hellman Parameter element is identical to the value it inserted in its (Re)Association Request frame. If it is not the STA shall fail (Re)Association. Otherwise it shall extract the AP’s public key from the Public Key field using the conversion defined in 11.3.7.2.5 (Octet string to element conversion). The extracted public key, ***N***, shall then be validated using the element validation technique from 11.3.5.4 (Processing of a peer’s SAE Commit message). If validation of the public key fails, the STA shall fail (Re)Association.

Otherwise, the STA shall then complete the Diffie-Hellman key exchange and generate a shared secret a PMK, and a PMKID:

 ***S*** = scalar-op(*m*, ***N***)

 *s* = F(***S***)

 *PMK* = KDF-Hash-256(s, “OWE PMK Generation”, F(***M***) || F(***N***))

 *PMKID* = Truncate-128(SHA-256(F(***M***) || F(***N***)))

Where KDF-Hash-256 is the key derivation function defined in 11.6.1.7.2 (Key derivation function (KDF)) using the hash algorithm defined by the AKM in Table 8-131 to generate a 256-bit key, and F() is the element-to-scalar mapping function from 11.3.4 (Finite Cyclic groups). Intermediate data, ***S*** and *s*, shall be irretrievably deleted upon generation of the PMK. Upon generation of the PMKID the public keys ***M*** and ***N*** can be deleted. A PMKSA shall be created using the PMK and PMKID.

**References:**