IEEE P802.11
Wireless LANs

|  |
| --- |
| Proposed Resolution for Assigned Security CIDs |
| Date: 2013-03-21 |
| Author(s): |
| Name | Affiliation | Address | Phone | email |
| René Struik | Struik Security Consultancy | Toronto ON | +1 415 690-7363+1 647 867-5658Skype: rstruik | rstruik.ext@gmail.com |

Abstract

This document presents suggested text to implement “large objects” with TGai, as indicated in 13/201r4. Suggested text is relative to 802.11-2012 and TGai/D0.4.

**3.1 Definitions**

***Insert the following new definition:***

**Conceptual object:** octet string that may have syntax or semantics that are only partially specified with 802.11 (e.g., higher-layer information). This conceptual object is only conveyed between a STA and AP, after representing this as an ordered sequence of information elements, whereas the receiving entity may only act on this object after converting this back to the corresponding conceptual object

**8.4 Management frame body components**

***Insert section 8.4.1a as indicated:***

**8.4.1a Conceptual objects**

**8.4.1a.1 General**

A conceptual object is an octet string that may have syntax or semantics that are only partially specified with 802.11 (e.g., higher-layer information). This conceptual object is only conveyed between a STA and AP, after representing this as an ordered sequence of information elements (8.4.1a.2.1), whereas the receiving entity may only act on this object after converting this back to the corresponding conceptual object (8.4.1a.2.2).

To facilitate description of the mapping from conceptual objects to sequences of information elements (8.4.1a.2.1) and the corresponding inverse mapping (8.4.1a.2.2), conceptual objects are described as “generalized information elements”, as follows.

Conceptual objects are defined to have a common general format consisting of a 1-octet Conceptual Object Class field and a variable-length object class-specific Information field. Each element is assigned a unique Object class ID as defined in this standard. The format of the Conceptual Object is shown in Fig. 8.4.1a-1.

|  |  |  |
| --- | --- | --- |
|  | Conceptual Object Class Id | Conceptual Object Value |
| Octets: | 1 | Variable |
| **Figure 8.4.1a – Conceptual object format(11s)** |  |  |

The set of valid object classes is defined in Table 8.4.1a-2.

***Editor note: Conceptual Object Classes 1-3 are based on TLV objects currently defined in 13/235r2 (Table, p. 6).***

|  |  |  |
| --- | --- | --- |
| **Conceptual Object** | **Conceptual Object Class Id** | **Description** |
| Higher-Layer | 0 | (see 8.4.1a.2) |
| FILS HLP Wrapped Data | 1 | (see 8.4.1a.3) |
| FILS IP Address Assignment | 2 | (see 8.4.1a.4) |
| FILS DNS Information | 3 | (see 8.4.1a.5) |
|  | 4-255 | Reserved |

**8.4.1a.2 Conversion scheme**

**8.4.1a.2.1 Representation of Conceptual Objects**

This involves the transformation of the Conceptual Object field to an ordered sequence of information element, according to the following stipulations:

* The Element Id of each such information element shall be set to the “Conceptual Object Segment IE”, as specified in Table 8-54;
* The Length field of each such information element shall have a value between 0 and 255 (inclusive);
* The right-concatenation of all Information Fields in the resulting ordered sequence of information elements shall be equal to the Conceptual Object field;
* The rightmost (last) information element in the resulting ordered sequence of information elements shall have its Length field set to zero.
* The representation of an ordered sequence of conceptual objects shall be defined as the ordered sequence of the outcomes of the transformation of each single conceptual object in this sequence, using this original ordering.

The procedure by which the partitioning of the Conceptual Object field over the information elements is determined is out of scope of this specification.

**8.4.1a.2.2 the Inverse Mapping**

This involves the transformation of an ordered sequence of information elements to an ordered sequence of Conceptual Object fields, according to the following stipulations:

* The transformation shall fail if not all information elements have their Element Id set to the “Conceptual Object Segment IE”, as specified in Table 8-54;
* The transformation shall partition the ordered sequence of information elements into ordered subsequences of information elements, where the rightmost (last) information element in each such ordered subsequence has a Length field set to zero and where each other information element in this subsequence has a Length field set to a nonzero value. If this partitioning is not possible, the transformation shall fail.
* The transformation shall convert each subsequence into a Conceptual Object field which is equal to the right-concatenation of all Information Fields in the ordered subsequence; the ordered sequence of Conceptual Object fields shall be defined as the result of ordering the Conceptual Object fields reconstructed from each subsequence, according to the partitioning order by which these subsequences were previously derived.

Note: It is easy to see that the composition of these two conversion function results in the original sequence of Conceptual Object fields (if one ignores empty octet strings).

**8.4.1a.3 Higher-Layer**

The higher-later conceptual object indicates a “true” higher-layer object. As such, this may be any octet string of length smaller than 0xffff, with syntax and semantics entirely out-of-scope of the standard.

**8.4.1a.4 FILS HLP Wrapped Data**

***Editorial note – see 13/235r2 (8.4.ai2.ai1), except for the 1-octet Type field (which is Conceptual Object Class Id above, in (8.4.1a.1)).***

**8.4.1a.5 FILS IP Address Assignment**

***Editorial note – see 13/235r2 (8.4.ai2.ai2), except for the 1-octet Type field (which is Conceptual Object Class Id above, in (8.4.1a.1)).***

**8.4.1a.6 FILS DNS Information**

***Editorial note – see 13/235r2 (8.4.ai2.ai3), except for the 1-octet Type field (which is Conceptual Object Class Id above, in (8.4.1a.1)).***

**8.4.2.1 General**

***Insert the following row/Information Elements to Table 8-54:***

|  |  |  |  |
| --- | --- | --- | --- |
| **Element** | **Element ID** | **Length of indicated element (in octets)**  | **Extensible** |
| Conceptual Object Segment (see x.x.x) | <ANA> | 2-257 | Representation of a segment of a conceptual object (if the length field of this element is set to a positive value), respectively representation of a separator between conceptual objects (if the length field of this element is set to zero)  |
| Encryption indicator (see x.x.x) | <ANA> | 4 | Indicator of the octet length of the frame segment directly following it that is encrypted |

***Editorial note – the Element Id for the Encryption Indicator should be set to the smallest available number no already specified in 802.11-2012.***

***Insert the following subclause:***

**8.4.2.187 Encryption Indicator**

The encryption indicator is used to uniquely indicate the frame segment that is encrypted. The format of the Encryption Indicator element is shown in Fig. 8.4.2.187-1.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Element Id | Length | Length Encrypted Segment |
| Octets: | 1 | 1 | 2 |

The Element Id shall be set to the “Encryption Indicator IE”, as specified in Table 8-54.

***Modify section D0.4/11.11.2.4 as indicated:***

**11.11.2.4 Key confirmation with FILS authentication**

Key confirmation for FILS Authentication is an Associate Request followed by an Associate Response. The

Association Request and Association Response shall be protected using the KEK2 according to 11.11.2.5 and 11.11.2.6.

Upon the completion of key establishment (11.11.2.2) and key derivation (11.11.2.3) the STA shall construct an 802.11 associate request frame indicating its selected ciphersuite and the FILS AKM, and the FILS Key Confirmation element. The content of the Key Auth field of the Key Confirmation element depends on the type of FILS authentication.

The AP transfers any necessary KDEs to the STA in the Association Response frame. The AP may include one or more KDEs using the FILS KDE container. The format and the rules for transferring the KDE shall follow 11.6.2 (EAPOL Key Frames).

For FILS Authentication using a trusted third party, the Key Auth field of the Key Confirmation element of the Association Request shall be:

Key-Auth = HMAC-SHA256(KCK2, NSTA | NAP | STA-MAC | AP-BSSID).

For FILS Authentication without a trusted third party, the Key Auth field of the Key Confirmation element in the Association Request shall contain a digital signature using the STA's private key, the specific construction of the digital signature depends on the crypto-system of the public/private key pair:

Key-Auth = Sig-STA(gSTA | gAP | NSTA | NAP | STA-MAC | AP-BSSID).

Where Sig-STA indicates a digital signature using the STA's private key, gSTA is the octet-string representation

of the STA's public Diffie-Hellman value, gAP is the octet-string representation of the AP's public

Diffie-Hellman value, NSTA is the nonce selected by the STA, and NAP is the nonce selected by the AP.

The 802.11 Association Request frame shall be secured as follows:

* The input key shall be the KEK2
* The plaintext indicator shall be a set of information elements contained in the Association Request frame that follow the FILS Session element. The procedure by which STA selects this set of information elements is outside scope of the specification. (We assume that the indicator does not “break” conceptual objects (8.4.1a))
* The input string shall be:
1. The STA MAC
2. The AP BSSID
3. The STA's nonce
4. The AP's nonce
5. The contents of the Association Request frame following the capability element (inclusive)
* The input key, the plaintext indicator, and the input string shall be passed to the encrypt-and-authenticate operation specified in 11.11.2.6.
* The output ciphertext from 11.11.2.6 shall become the remainder of the Association Request frame that follows the FILS Session element.

The resulting 802.11 Association Request frame shall be transmitted to the AP.

The received 802.11 Association Request frame shall be processed as follows:

* The input key shall be the KEK2
* The input string shall be:
1. The STA MAC
2. The AP BSSID
3. The STA's nonce
4. The AP's nonce
5. The contents of the Association Request frame that follow the capability element (inclusive)
* The input keys and the input string shall be passed to the decrypt-and-verify operation specified in 11.11.2.7.

If the output from 11.11.2.7 returns a failure, authentication shall be deemed a failure. Otherwise, the Key-Auth from the decrypted Association frame shall be checked. If it is incorrect, authentication shall be deemed a failure. If authentication is deemed a failure, the KCK2, KEK2, KCK, KEK, and TK shall be irretrievably destroyed. If authentication is not deemed a failure, the AP shall check the Key-Auth field in the Key Confirmation element.

For FILS Authentication using a trusted third party, the AP shall construct a verifier as follows:

Key-Auth' = HMAC-SHA256(KCK, NSTA | NAP | STA-MAC | AP-BSSID)

If Key-Auth' differs from the Key-Auth field in the Key Confirmation element, authentication shall be deemed a failure.

For FILS Authentication without a trusted third party, the AP shall use the STA's (certified) public key from the FILS Public Key element in the Association frame to verify the contents of the Key-Auth field of the Key Confirmation element. The specific technique for verification depends on the crypto-system used by the public key. If verification fails, authentication shall be deemed a failure.

If authentication is a failure, the KCK2, KEK2, KCK, KEK, and TK shall be irretrievably destroyed. Otherwise, the AP shall then construct an 802.11 associate response frame confirming the selected ciphersuite and the FILS AKM, and containing the FILS KDE Container, and its own Key-Auth.

For FILS authentication using a trusted third party, the Key Auth field of the Key Confirmation element in the Association Response shall be:

Key-Auth = HMAC-SHA256(KCK2, NAP | NSTA | AP-BSSID | STA-MAC).

For FILS Authentication without a trusted third party, the Key Auth field of the Key Confirmation element in the Association Response shall contain a digital signature using the AP's private key, the specific construction of the digital signature depends on the crypto-system of the public/private keypair:

Key-Auth = Sig-AP(gAP | gSTA | NAP | NSTA | AP-BSSID | STA-MAC ).

Where Sig-AP indicates a digital signature using the AP's private key, and where gSTA, gAP, NSTA, and NAP are the same as in the construction of the Association Request.

The 802.11 Association Response frame shall be protected as follows:

* The input keys shall be the KEK2
* The plaintext indicator shall be a set of information elements contained in the Association Request frame that follow the FILS Session element. The procedure by which AP selects this set of information elements is outside scope of the specification. (We assume that the indicator does not “break” conceptual objects (8.4.1a))
* The input string shall be
1. The AP BSSID
2. The STA MAC
3. The AP's nonce
4. The STA's nonce
5. The contents of the Association Response frame that follow the capability element (inclusive)
* The input keys, the plaintext indicator, and the input string shall be passed to the encrypt-and-authentication operation specified in 11.11.2.6.
* The output ciphertext shall become the remainder of the Association Response frame that follows the FILS Session element.

The resulting 802.11 Association Response frame shall be transmitted to the STA.

The STA shall process the received 802.11 Association Response frame as follows:

* The input key shall be the KEK2
* The input string shall be:
1. The AP BSSID
2. The STA MAC
3. The AP's nonce
4. The STA's nonce
5. The contents of the Association Response frame following the capability element (inclusive)

The input keys and the input string shall be passed to the decrypt-and-verify operation specified in 11.11.2.7.

If the output from 11.11.2.7 returns failure, authentication shall be deemed a failure. Otherwise, the Key-Auth from the decrypted Authentication frame shall be checked. If it is incorrect, authentication shall be deemed a failure. If authentication is deemed a failure, the KCK2, KEK2, KCK, KEK, and TK shall be irretrievably destroyed. If authentication is not deemed a failure, the AP shall check the Key-Auth field in the Key Confirmation element.

For FILS Authentication using a trusted third party, the STA shall construct a verifier as follows:

Key-Auth' = HMAC-SHA256(KCK2, NAP | NSTA | AP-BSSID | STA-MAC).

If Key-Auth' differs from the Key-Auth field in the Key Confirmation element, authentication shall be deemed a failure .

For FILS Authentication without a trusted third party, the STA shall use the AP's (certified) public key from the FILS Public Key element in the Association frame to verify the contents of the Key-Auth field of the Key Confirmation element. The specific technique for verification depends on the crypto-system used by the public key. If verification fails, authentication shall be deemed a failure.

If authentication is a failure, the KCK2, KEK2, KCK, KEK, PMK, and TK shall be irretrievably destroyed.

Otherwise authentication succeeds. In that case, STA and AP shall irretrievably destroy the temporary keys

KCK2 and KEK2 and both shall use the TK with the cipher indicated by the negotiated. The KCK, KEK, and PMK shall be used for subsequent key management as specified in clause 11.5. The STA and AP shall set the lifetime of the PMKSA to the value dot11RSNAConfigPMKLifetime.

***Modify section D0.4/11.11.2.6 as indicated:***

**11.11.2.6 Encrypt and authenticate operation for FILS association frames**

**11.11.2.6.1 Input Transformation (Re-order Plaintext if Applicable)**

* Determine the total octet length of the plaintext from the plaintext indicator and set the Information field of the Encryption Indicator element to this value.
* Parse the input string and re-order IEs so that the plaintext indicated by the plaintext indicator occupies the smallest contiguous frame segment possible (thereby not reordering IEs with the same Element ID).
* Partition the resulting string as A1 || P || A2, where the string P corresponds to the plaintext indicated by the plaintext indicator (which contains all IEs with A1 and A2 in ascending order).

**11.11.2.6.2 Encrypt and authenticate operation**

The AEAD scheme of 11.11.2.5 shall be used with the 802.11 Associate Request frame (for enciphering by

STA) or with the 802.11 Associate Response frame (for enciphering by AP), with the following instantiation:

* The key *K* shall be set to KEK2;
* The associated data string shall be set to the right-concatenation of the strings A1 and A2 determined in 11.11.2.6.1;
* The string *P* shall be set to the string determined in 11.11.2.6.1;
* The nonce *N* shall be set to
1. For processing by STA: use the 13-octet all-zero string;
2. For processing by AP: use the 13-octet all-one string.

If the encryption-authentication process is unsuccessful, output a failure; otherwise, output the string A1 || C || A2.

**11.11.2.7.3 Output Transformation (Indicate the Associated Data and Ciphertext)**

* Substitute the input string by the string A1 || LBL || C || A2 (which contains all visible IEs in ascending order). Here, LBL is equal to the Encryption Indicator Element.

***Modify section D0.4/11.11.2.7 as indicated:***

**11.11.2.7 Decrypt and verify operation for FILS association frames**

**11.11.2.7.1 Input Transformation (Determining the Associated Data and Ciphertext)**

* Parse the input string to determine the Security Indicator Element. If this element is not found, the procedure shall output a failure.
* Partition the input string as A1 || C || A2, where the octet string C has as length the value contained in the Encryption Indicator Information field and where the rightmost 4 octets of the string A1 are equal to the Encryption Indicator Element. Subsequently remove these 4 rightmost octets of the string A1. If this partitioning is not possible, the procedure shall output a failure.

**11.11.2.7.2 Decrypt and Verify Operation**

The AEAD scheme of 11.11.2.5 shall be used with the 802.11 Associate Request frame (for deciphering by

STA) or with the 802.11 Associate Response frame (for deciphering by AP), with the following instantiation:

* The key K shall be set to KEK2;
* The associated data string shall be set to the right-concatenation of the strings A1 and A2 determined in 11.11.2.7.1;
* The string C shall be set to the string C determined in 11.11.2.7.1;
* The nonce N shall be set to
1. For processing by AP: use the 13-octet all-zero string.
2. For processing by STA: use the 13-octet all-one string;

The function shall output the string A1 || P || A2 if the decryption -verification process is successful and shall output a failure otherwise.

**11.11.2.7.3 Output Transformation (Re-order Recovered Plaintext if Applicable)**

* Parse the output string A1 || P || A2 and re-order IEs that are out of order (thereby not reordering IEs with the same Element ID).

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |
| --- |
|  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |
| --- |
|  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |
| --- |
|  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |
| --- |
|  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |

|  |  |
| --- | --- |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
|  |

|  |  |
| --- | --- |
|  |  |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |

|  |
| --- |
|  |
|  |  |  |
|  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Substitute the input string by the result of this transformation (which contains all IEs in ascending order).