|  |
| --- |
| Proposed Resolution for Some Security CIDs |
| Date: 2014-03-11 |
| 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 discusses some security-related LB#198 comments related to D1.0 of the TGai specification.

**Summary sheet: Suggested resolution of a selection of comments from 13/1076r30:**

*Comment disposition below (in black) was already discussed in 13/1193r1, posted September 17, 2013)*

|  |  |  |
| --- | --- | --- |
| **CID#** | **Resolution** | **Brief rationale** |
| 2456 | Accept in Principle | Editorial glitch |
| 2897 | Accept | Editorial glitch |
| 2202 | Reject | Comment on nonce use, already discussed at length with D0.5 |
| 3002 | Reject | Comment on nonce use, already discussed at length with D0.5 |
| 3154 | Reject | Duplicate of #3002 |
| 3193 | Reject | Duplicate of #3002 |
| 3089 | Reject | Comment on nonce use, already discussed at length with D0.5. Commenter already gets extensibility he wants, fragmentation irrelevant |
| 2201 | Accept in Principle | Small textual changes to support GCM mode, currently in *draft* 802.11ac |
| 3003 | Accept in Principle | Virtually same as #2201 |
| 3155 | Accept in Principle | Duplicate of #3003 |
| 3194 | Accept in Principle | Duplicate of #3003 |
| 2986 | Accept | Add cross-reference to section on key usage in 802.11-2012 |
| 3245 | Accept in Principle | Eradicate all FILS re-association language |
| 2204 | Reject | Keys used for different purpose should be logically separated. |
| 2203 | Accept in Principle | Specify that AAD string is formed by right-concatenation of its elements |
| 3259 | Accept in Principle | Specify even more clearly that keys should be destroyed once not needed |
| 3244 | Accept in Principle | Addressed by eradicating FILS re-association language (see also CID#3245) |
| 3243 | Accept | Editorial glitch |
| 2987 | Reject | Keys are already uniquely specified, as substrings of kdf output |
| 2997 | Reject | Simple function call, with parameter instantiation (isn’t this obvious?) |
| 2996 | Reject | Duplicate of #2997, except for text location |
| 2995 | Reject | Standard term, no alternative suggested; besides, this is purely editorial |
| 2877 | Accept in Principle | Editorial glitches; need to clarify 11.11.2.8 (see also #2222) |
| 2876 | Accept | Editorial glitch |
| 2875 | Accept in Principle | Suggest to use option #1 in comment (stay mute when authentication failure). This seems to be closest to silently dropping corrupted frame. |
| 2853 | Accept | Fix errors in state diagram Fig. 10-6 (also one not in comment) |

**DETAILS: Suggested resolution of a selection of comments from 13/1076r30:**

**CID #2456:** (George Cherian) 10, p. 84, l. 64:

Disassociation should change FILS's state to stae-1.

Suggested resolution: Accept with change.

Motivation: The state diagram transition should be towards State 2 (this would have been to State 1 with de-authentication).

**CID #2897:** (Lin Cai), p. 84, l. 64:

There is no 'four subfields' defined in the context, but only three subfields as in Figure 8-401dr.

Change 'four' to 'three'.

Suggested resolution: Accept.

Motivation: obvious editorial glitch. BTW – This figure is not in Draft D1.3 any more.

**CID #2202:** (Dan Harkins) 11.11.2.7, p. 107, l. 32:

Do not use fixed nonces in AEAD schemes. Adopt the AEAD scheme from 11-13/0806r2.

Suggested resolution: Reject.
Motivation: The security of the CCM mode of operation requires that nonces are not reused with the same key. This is indeed the case with Draft D1.0 (as this was with Draft D0.5, when this comment was already discussed in great detail). This also fully conforms to the referenced NIST specification (see Section 5.3 of NIST SP 800-38C).
Further explanation for the commenter: Draft D1.0 only introduces one new use of the CCM mode of operation (in Clause 11.11.2.5) and this mode is invoked precisely once in each direction during key confirmation (Clause 11.11.2.4): once from STA to AP (Association Request) and once from AP to STA (Association Response). Since the two key confirmation messages use different nonces and the key KEK2 is only used during key confirmation and "freshly" generated at each invocation of the FILS protocol (in Clause 11.11.2.3), nonce reuse does not occur. Please note that the nonces used by either party are in fact counters, where STA uses as counters the even integers counting up from zero and where AP uses as counters the odd integers counting down. Since each side only uses the counter once, this comes down to using the counters 0 and 2104-1, respectively. As a final note, key confirmation messages remain unsecured frames in the sense of 802.11-2012, so are not impacted by 802.11-2012 frame fragmentation (as defined in Clause 9.2.7 hereof).

**CID #3002:** (Michael Montemurro) 11.11.2.7, p. 107, l. 32:

Fixed nonces, mistake.

Adopt AEAD scheme in 11-13/0806r2

Suggested resolution: Reject.
Motivation: see CID #2202.

**CID #3154:** (Richard Kennedy) 11.11.2.7, p. 107, l. 32:

Fixed nonces, mistake.

Adopt AEAD scheme in 11-13/0806r2

Suggested resolution: Reject.
Motivation: see CID #3154, of which this comment is a duplicate.

**CID #3193:** (Roger Durand) 11.11.2.7, p. 107, l. 32:

Fixed nonces, mistake.

Adopt AEAD scheme in 11-13/0806r2

Suggested resolution: Reject.
Motivation: see CID #3154, of which this comment is a duplicate.

**CID #3089:** (Paul Lambert) 11.11.2.5, p. 106, l. 47:

The usage of the CCM algorithm prevents any future extensions or fragmentation approaches in this part of IEEE 802.11. Adopt solution in .: IEEE 802.11-13/0806r2 or deine the usage of the nonce values as counters

Suggested resolution: Reject.
Motivation: see CID #2202. In particular, please note that key confirmation messages remain unsecured frames in the sense of 802.11-2012, so are not impacted by frame fragmentation and that the nonce values are already counters (where each counter is only used once).

**CID #2201:** (Dan Harkins) 11.11.2.5, p. 106, l. 47:

11ai follows 11ac, it should take into account changes that will be made to the standard by the time it passes sponsor ballot

No, AES-CCM is not the "only such scheme specified". Furthermore, just saying "AES-CCM" is ambiguous. If the first sentence is true then the second sentence should be removed. And when you remove it, get rid of "the following instantiation" too.

Suggested resolution: Modified resolution (accept in principle):

Motivation: While this draft is an amendment based on IEEE 802.11mc/D1.0, IEEE P802.11ad-2012., IEEE P802.11ac/D5.0, and IEEE P802.11af/D5.0, officially the draft is an amendment of 802.11-2012, which only specifies CCM mode. This being said, it is expected that the GCM mode that is in Draft 802.11ac/D5.0, will ultimately make it as part of the standard.

Suggested to change 11.11.2.5, 11.11.2.6, 11.11.2.7 as follows:

***Clause 11.11.2.5 AEAD scheme***

*Change line 48 as follows:*

Change “the only such scheme specified is” to “the only such scheme specified with 802.11-2012”.

*Add the following paragraph at the end of 11.11.2.5:*

The draft specification 802.11ac/D5.0 also specifies the AES-GCM mode of operation, which is the GCM scheme specified in NIST SP 800-38D with the following instantiation:

* The block cipher shall be AES-128 (see FIPS Pub 197);
* The authentication tag length (parameter t) shall be set to t=16;
* The length of the nonce (called IV) shall be set to 96 bits.

Either mode of operation can be used, as indicated by the cipher suite.

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

*Change l. 7-9 as follows:*

Remove the qualifier “13-octet”.

(Note: thus, with CCM mode, one will use 13-octet nonces; with GCM mode, one will use 12-octet nonces.)

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

*Change l. 31-33 as follows:*

Remove the qualifier “13-octet”.

(Note: thus, with CCM mode, one will use 13-octet nonces; with GCM mode, one will use 12-octet nonces.)

**CID #3003:** (Michael Montemurro) 11.11.2.5, p. 106, l. 47:

AES-CCM, only scheme specified.

This is false, use wording that is agnostic and expandable.

Suggested resolution: Modified resolution (accept in principle).

Motivation: see CID #2201.

**CID #3155:** (Richard Kennedy) 11.11.2.5, p. 106, l. 47:

AES-CCM, only scheme specified.

This is false, use wording that is agnostic and expandable.

Suggested resolution: Modified resolution (accept in principle).

Motivation: see CID #3003, of which this comment is a duplicate.

**CID #3194:** (Roger Durand) 11.11.2.5, p. 106, l. 47:

AES-CCM, only scheme specified.

This is false, use wording that is agnostic and expandable.

Suggested resolution: Modified resolution (accept in principle).

Motivation: see CID #3003, of which this comment is a duplicate.

**CID #2986:** (Michael Montemurro) 11.11.2.3, p. 103, l. 31:

Circular reference "...shall be used in exactly the same way as same-named keys of IEEE 802.11-2012 (but now derived as specified above)."

Point to the specific clauses in IEEE 802.11-2012 that this sentence references.

Suggested resolution: Accept.

*Change l. 32 as follows:*

Change “of IEEE 802.11-2012” to “of IEEE 802.11-2012 (see 4.10.4.2)”.

**CID #3245:** (Santosh Ghanshyam Pandey) 11.11.2.6, p. 106, l. 60:

It would be dangerous for at least the KEK2 key to be used more than one time, because of the fixed Nonce values used for AES-CCM. When this is used for an Association Request/Response a part of FILS Authentication (as shown in Figure 4-21a), this seems safe enough. But the FILS Secure Container is also present in Reassociation Request and Response frames (see Clauses 8.3.3.7 and 8.3.3.8), so presumably the process described in Clause 11.11.2.6 is used to protect those frames. It is not clear how the KEK2 is used safely in that case.

Clarify how Encrypt and authenticate works for Reassociation Request and Response frames, as well as Decrypt and verify in clause 11.11.2.7.

Suggested resolution: Modified resolution.

Motivation: Both keys KCK2 and KEK2 are strictly temporary and are only used during key confirmation, as part of the FILS authentication protocol (see 11.11.2.3, l. 26-29). Thus, Clause 11 of D1.0 does not apply to any other protocol, so nor with Reassociation.

In fact, isn’t the entire text surrounding FILS Reassociation in error and shouldn’t it be killed off altogether here? After all: the whole point of FILS authentication is that now the entire authenticated key establishment procedure takes only four protocol flows (rather than the umpteen with 802.11-2012, which at the time may have motivated use of Reassociation as a “short-cut” for going through Authentication and Association again), so if one needs to reassociate, one should simply execute another run of the (four-pass) FILS protocol.

Suggested changes to 8.3.3.6, 8.3.3.7, 8.4.2.186.1, 8.4.2.186.2, 8.4.2.186.3, 8.4.2.186.4, 10.3.2, 10.3.5.1, 10.44.4, 10.44.4.1, 10.44.4.2

{Remove FILS reassociation altogether}

*Remove Clause 8.3.3.7 entirely.*

*Remove Clause 8.3.3.8 entirely.*

*Change Clause 8.4.2.186.1 as follows:*

Remove all occurrences of “Reassociation Request” or “Reassociation Response”.

*Change Clause 8.4.2.186.2 as follows:*

Remove all occurrences of “Reassociation Request” or “Reassociation Response”.

*Change Clause 8.4.2.186.3 as follows:*

Remove all occurrences of “Reassociation Request” or “Reassociation Response”.

*Change Clause 8.4.2.186.4 as follows:*

Remove all occurrences of “Reassociation Request” or “Reassociation Response”.

*Change Clause 10.3.2 as follows:*

Remove “Reassociation” from the state diagram portions relevant to FILS (State 5 to State 4 transition and vice-versa) in Fig. 10-6.

*Change Clause 10.3.5.1 as follows:*

Remove the sentence on l. 58-59 entirely (i.e., remove “Successful reassociation sets a FILS STA's state to State 4 and enables it to exchange Class 3 frames”).

*Change Clause 10.44.4 as follows:*

Remove all occurrences of “Reassociation Request” or “Reassociation Response” (including the one in the title).

*Change Clause 10.44.4.1 as follows:*

Remove all occurrences of “Reassociation Request” or “Reassociation Response”.

*Change Clause 10.44.4.2 as follows:*

Remove all occurrences of “Reassociation Request” or “Reassociation Response”.

**CID #2204:** (Dan Harkins) 11.11.2.3, p. 102, l. 14:

KEK2 and KCK2 are superfluous

If KEK and KCK are used for encryption and integrity protection then there is no need for KEK2 and KCK2. Get rid of them and just use KEK and KCK

Suggested resolution: Reject.

Motivation: The keys KCK2 and KEK2 are temporary keys and used during key confirmation only. Since key confirmation messages remain unsecured frames in the sense of 802.11-2012, these keys should be logically separated from the keys KCK and KEK that are used to secure ordinary 802.11-2012 frames.

**CID #2203:** (Dan Harkins) 11.11.2.4, p. 104, l. 17:

How does one represent this set as a single datum?

Explain what one does with those things that the "AAD shall be". Do I hash them all together? Are they a vector of inputs to the scheme from 11-13/0806r2? This needs to be fixed in all references to AAD in 11.11.2.4, not just the given page and line numbers (which this spreadsheet limited to a single number).

Suggested resolution: Modified (Accept in principle)

Motivation: this is indeed ambiguous. Suggest to specify the corresponding strings as right-concatenation of the listed elements, in order.

Suggested changes to 11.11.2.4:

*Change p. 104, l. 15 as follows:*

Change “The input AAD shall be:” to “The input AAD shall be the right-concatenation of the strings”.

*Change p. 104, l. 31 as follows:*

Change “The input AAD shall be:” to “The input AAD shall be the right-concatenation of the strings”.

*Change p. 105, l. 39 as follows:*

Change “The input AAD shall be:” to “The input AAD shall be the right-concatenation of the strings”.

*Change p. 105, l. 65 as follows:*

Change “The input AAD shall be:” to “The input AAD shall be the right-concatenation of the strings”.

**CID #3259:** (Rob Sun) 11.11.2.3, p. 103, l. 12:

How and where to store KCK2/KEK2 which should be transient and only for FILS key confirmation? It needs to define when to destroy the KCK2/KEK2. The assumption is PMKSA which is lack of definition in elements for FILS would store the KCK2/KEK2 which may not be ideal

Please clarify

Suggested resolution: Modified (Accept in Principle).

Motivation: This is an implementation question, not a specification question. The keys KCK2 and KEK2 are temporary keys, only to be used during key confirmation. According to the last paragraph of 11.11.2.4, these keys are destroyed in case either STA or AP concludes that the FILS protocol was successful. Moreover, the draft also stipulates to destroy these keys in case either party concludes that the FILS protocol is not successful (see, e.g., p. 106, l. 16 and l. 36, for STA, and p. 104, l. 58, and p. 105, l. 11 for AP). Nevertheless, if this helps, one could add yet another reminder in 11.11.2.3.

*Change 11.11.2.3 as follows:*

Add the following sentence at the end of l 28: Both keys KCK2 and KEK2 shall be irretrievably destroyed as soon as these are not needed any more during key confirmation.

**CID #3244:** (Santosh Ghanshyam Pandey) 11.11.2.3, p. 103, l. 27:

This clause states that the KCK2 and KEK2 are temporary, which implies that they are used for only the Association Request and Response as described in clause 11.11.2.4. But there is not guidance given as to whether these keys are used for the Reassociation Request and Response frames.

Clarify how keys are generated when used with the Reassociation Request and Response frames.

Suggested resolution: Modified (Accept in Principle).

Motivation: The notation that FILS Reassociation frames exist seems to be flawed. See also resolution of CID#3245, which suggested to kill this off entirely, thereby also addressing this comment.

**CID #3243:** (Santosh Ghanshyam Pandey) 11.11.2.3, p. 103, l. 27:

The reference for encrypt-and-authentication should be 11.11.2.6, and decrypt-and-verify should be 11.11.2.7.

Change as per the comment

Suggested resolution: Accept.

Motivation: Editorial glitch.

**CID #2987:** (Michael Montemurro) 11.11.2.3, p. 103, l. 13:

Where is the description of how the keys are derived from the KDF output.

Eiher explicity show the Key derivations from the KDF output or point ot the existing sub-clauses where these are defined.

Suggested resolution: Reject.

Motivation: The keys are specified as substrings, in order, of the corresponding kdf-output and uniquely defined by their length.

**CID #2997:** (Michael Montemurro) 11.11.2.6, p. 107, l. 25:

"The associated data string A shall be set to the string AAD". What does this mean?

Reword the sentence to describe exactly the context and string are set to.

Suggested resolution: Reject.

Motivation: This is simply a function call, where input parameter A is set to the value AAD (of the same type).

**CID #2996:** (Michael Montemurro) 11.11.2.6, p. 107, l. 1:

"The associated data string A shall be set to the string AAD". What does this mean?

Reword the sentence to describe exactly the context and string are set to.

Suggested resolution: Reject.

Motivation: See CID#2997 (of which this is a duplicate, except for location in text).

**CID #2995:** (Michael Montemurro) 11.11.2.5, p. 106, l. 45:

What is a "scheme"? What does this mean?

Suggested resolution: Reject.

Motivation: No suggestion provided for rewording of standard security term. Besides, this cannot possibly be a technical comment.

**CID #2222:** (David Goodall) 11.11.2.8, p. 107:

The description of what is encrypted and decrypted is in conflict with section 11.11.2.4 which indicates that the portion of the association frames after the FILS session element is subject to encrypt/decrypt.

Resolve the conflict with section 11.11.2.4.

Suggested resolution: Modified (Accept in Principle.)

Motivation: The intention is that the container object (since piggy-backed info) is part of the to-be-encrypted-and-authenticated data in the key confirmation message. Admittedly, the current text is awkward, seems could be interpreted as suggesting that the container itself should be secured first before securing the key confirmation message itself.

*Suggested change to 11.11.2.8:*

Remove this clause entirely (since it is evident from the remainder of the draft that Association Response frame may include “piggy-backed info”.

[If people believe this is not clear from the draft, one needs to spell this out, but avoid giving the wrong impression about double encryption etc.]

**CID #2220:** (David Goodall) 10.3.3, p. 82, l. 65:

A STA in state 5 can also transmit class 2 frames.

Change line 65 to "A STA shall not transmit Class 2 frames unless in State 2 or State 3 or State 4 or State 5."

Suggested resolution: Accept.

Motivation: obvious glitch, in view of Fig. 10-6.

**CID #2877:** (Lei Wang) 11.11.2.5, p. 106, l. 43:

There seem multiple issues/questions with the Subsections 11.11.2.5 to 11.11.2.8, e.g.,

1) 11.11.2.6 and 11.11.2.7 are all the same text, except the very last paragraph. Is this supposed to be?

2).11.11.2.7 and 11.11.2.8 have exactly the same section title but different contents.

Make the following changes:

1) change the sentence in line 59 page 106 to the following:

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 Association Response frame (for enciphering by AP), with the following instantiation:

2). Change the Title of 11.11.2.8 to the following

Decrypt and verify operation for FILS Secure Container Element

Suggested resolution: Accept.

Motivation: Indeed, there seems to be an editorial glitch at the first paragraph of Clause 11.11.2.6, where “deciphering” should have read “enciphering” (2x in total). Moreover, the title of 11.11.2.8 is in error as well: this clause refers to wrapping and unwrapping of a container by AP and STA, respectively. However, since secure containers (piggy-backed info) is already part of the Association Response frame, their securing/unsecuring operation is already described in 11.11.2.4.

[If people believe this is not clear from the draft, one needs to spell this out, but avoid giving the wrong impression about double encryption etc.]

**CID #2876:** (Lei Wang) 11.11.2.5, p. 106, l. 17:

In the sentence in line 17 page 106, it should be about "STA" to process the Key Confirmation element in the received Association Response frame, not AP. Note that the similar processing for AP is specified in line 59 page 104.

Change the sentence in line 17 page 106 to the following

If authentication is not deemed a failure, the STA shall check the Key-Auth field in the Key Confirmation element.

Suggested resolution: Accept.

Motivation: obvious editorial glitch.

**CID #2875:** (Lei Wang) 11.11.2.5, p. 104, l. 55:

In section 11.11.2.4, Key confirmation for FILS authentication, there are multiple cases that "authentication shall be deemed a failure".

However, the current spec does not have any text specifying what the AP should do in this case. Note that the Key confirmation for FILS authentication uses Association request / response frames. There may be multiple choices, e.g.,

1) AP does nothing, then leave the STA to be timeout for waiting for the Association Response; then the SAT follows the existing procedure to handle such a timeout;

2) AP sends an Association Response frame with status code setting to a proper value as defined in Table 8-42 in 802.11mc/D1.5. If there is no suitable value, then a corresponding status code should be added.

propose to use 2).

Suggested resolution: Modified (accept in principle).

Motivation: From a security perspective, it seems better to remain mute when an association failure occurs and just let this time out. This seems to be in line with silently dropping corrupted or ill-formatted frames with ordinary 802.11 traffic. However, not sure what 802.11’s preferred approach is.

**CID #2853:** (Lei Wang) 10.3.2, p. 80, l. 26:

In Figure 10-6, on the transition from State 4 to State 5, the condition "1) unsuccesful (Re)Association (non-AP FILS STA)" is not correct and should be removed, because STAs in State 4 are after succuessfully associated, i.e., if the (re)Assoication is not succesful, the STA won't be in State 4.

remove the text "1) unsuccesful (Re)Association (non-AP FILS STA); 2)" from the text box above the arrowed line from State 4 to State 5 in Figure 10-6.

Suggested resolution: Modified (accept in principle).

Motivation: This state transition is indeed in error, so should be removed. In fact, one should add a state transition from State 5 of Fig. 10-6 to State 1 in case FILS Association fails (since all FILS authentication state will be destroyed).