IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Disposition of some SAE comments from LB236 and some comments made outside of LB236 | | | | |
| Date: 2019-03-08 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Dan Harkins | HPE | 3333 Scott boulevard  Santa Clara, California,  United States of America | +1 888 555 1212 |  |
|  |  |  |  |  |

Abstract

This submission addresses some SAE comments.

**CID 2276: SAE test vectors**

“According to John Van Boxtel: SAE test vectors' KDF (in Annex J.10) has 1-byte ‘i', it should be two-bytes.”

Discussion: I have no idea what John Van Boxtel is talking about. There is no “i” in J.10. There is no “i” in the protocol definition in section 12.4.

Proposed resolution: Reject. There is no “i” in J.10. There is no “i” in the protocol definition in section 12.4.

**CIDs 2590 and 2690: Privacy implications of password identifiers**

CID 2590: “The Password Identifier element is included in the unprotected authentication frame. It may violate the privacy of users (household). For example, it exposes a group of devices and number of devices that are sharing the same password. Particularly, when these devices belongs to the same household (apartment) in an apartment building, it violates the privacy of users/residents.” With a proposed change of “Delete the referenced subclause.”

And

CID 2690: “In SAE when Password Identifier is used, STA sends Password Identifier in the clear in Auth frame. Since the Password Identifier is typically the identifier of a long-term password, the same Password Identifier would be sent each time a STA performs SAE authentication with a given network, and this could be used by bad actors to track a user's location with privacy implications.” With a proposed change of “Provide means by which, e.g. during each SAE auth/assoc procedure, the AP can securely provision STA with a randomly-generated pseudonym for the SAE Password Identifier which the STA uses on the next SAE auth/assoc with that AP.”

Discussion: Glad to see people showing an interest in client privacy!

Using a credential from the dot11RSNAConfigPasswordValueTable whose dot11RSNAConfigPasswordPeerMac is the broadcast MAC address also leaks privacy information, namely membership in the group who knows this password. An SAE password identifier similarly identifies membership in a group and the size of that group may be but cannot be expected to be one. The smaller the group, the greater the privacy implications. If the members of the group identified by an SAE password identifier is, in fact, one then it can be used to track this singular user much in the same way that using the STA’s MAC in the dot11RSNAConfigPasswordPeerMac (a perfectly valid thing to do) does, and in much the same way as a STA using a fixed MAC address does regardless of authentication protocols.

To suggest deletion of password identifiers (CID 2590) as a solution to the privacy issue is to remove a useful feature because some might not want to use it. It is not mandatory to use so if one is alarmed at the privacy implications of a cleartext password identifier then one can choose to forgo the benefits that password identifiers afford. But to deny everyone the ability to use password identifiers simply because some might not want to use them is to open up the standard to something akin to the “heckler’s veto” where useful features would be removed—perhaps the use of fixed MAC addresses!—because some people might object to them or they could be misused. Removing a feature that has been implemented, and this feature has been implemented, would set a very bad precedent. Furthermore, satisfying the commenter by deleting the referenced subclause would leave references to password identifiers in other parts of the standard which would be confusing and form comment bait for future letter ballots.

Authentication requires an identity. Obfuscating the identity with a pseudonym (CID 2690) is an interesting idea. But managing pseudonyms would become increasingly difficult as the number of users sharing the password identifier grows. It would require a certain amount of synchrony among the members of the group or require an inordinate amount of memory to be retained by the AP to ensure that older pseudonyms are retained for users who do not associate as frequently as others will still generating new pseudonyms for those users that associate more frequently. The proposal to provide a randomly-generated pseudonym is interesting but should probably be handled at a higher layer outside the MAC.

Proposed resolutions:

CID 2590: Reject. The feature is useful and not mandatory to use. Password identifiers do not create any new vector of attack, they merely move an existing vector to another area. There are no new attacks possible with the use of password identifiers in SAE.

CID 2690: TBD

**CID 2546: Parsing of options in an SAE commit message**

“The approach used does not work, because the Scalar and Anti-Clogging Token fields are not elements, and hence could contain octet sequences that make them look like FFE, Finite Cyclic Group or Vendor Specific elements.” With a proposed change of “Make the Scalar and Anti-Clogging fields self-identifying by making them elements, not fixed fields.”

Discussion: “[M]ake them look” is vague and suggesting this vagueness causes a problem is completely hypothetical. It is not clear how the Scalar can be “[made] to look like... [a] Finite Cyclic Group or Vendor Specific elements.” Satisfying the commentor would make every existing implementation of SAE be incorrect which makes the proposed change a non-starter.

Furthermore, the existence of two independently developed and interoperable implementations of SAE that operate according to the Note is proof that the contention in the CID that “[t]he approach used does not work” is incorrect.

Proposed resolution:

Reject. The approach used is based on the lengths of components and not on their contents or how the contents “look”. Multiple independent and interoperable implementations using this approach exist. The proposed change would make all existing implementations of SAE be non-compliant.

**CID 2545: Normative word in Informative Note**

“’may be optionally’ -- no normative statements in NOTEs” with a proposed change of “Change the cited text at the referenced location to ‘can be’”.

Discussion: OK

Proposed resolution:

Accept

**CIDs 2529, 2530, 2532: No such stuff**

All of these CIDs note that some reference is inconsistent—“group field” not “finite cyclic group field”, etc. The proposed resolutions all say something like “change throughout the referenced subclause”.

Discussion: These CIDs all point to valid inconsistencies that should be fixed but it is unreasonable to put such a vague resolution in the hands of anyone other than the commenter who discovered all the instances in the referenced subclauses.

Proposed resolution: Assign to Mark Rison with “submission required”.

**CID 2527: Optional fields in an SAE Commit message**

“’An SAE Commit message consists of a scalar and an element’ -- it can also have a token, an identifier and a group” with a proposed change of “Change the cited text at the referenced location to ‘An SAE Commit message consists of a scalar, an element and optional a token, an identifer and/or a group’.”

Discussion: The words in question are part of a larger sentence and the CID leaves out important context. The sentence is “An SAE Commit message consists of a scalar and an element that shall be produced using the PWE and secrets generated in 12.4.5.2 (PWE and secret generation), as follows:” and a formula follows. So while it is undoubtably true that an SAE Commit message can also have a token, and identifier, and a group, those components are, at the particular time in the protocol being described, unimportant and have no bearing on the generation of the scalar and element which is what is being discussed in the cited text.

Proposed resolution:

Revised. Change the cited text to “The scalar and element in an SAE Commit message shall be produced using the PWE and secrets generated in 12.4.5.2 (PWE and secret generation), as follows:”.

**CID 2386: What’s the confirm?**

“It is not clear what the confirm object is/what is done with it” with a proposed change of “At line 8 after ‘A peer generates an SAE Confirm message’ add ‘confirm’ (without quotes) in italics.”

Discussion: The string being generated is the confirmation that is part of the SAE Confirm message. The subsequent subsection discusses extracting the peer’s confirmation from the received SAE Confirm message so it makes sense to add some text about inserting a confirmation into the SAE Confirm message.

Proposed resolution: Revised. See resolution in [this document].

*Instruct the editor to modify section 12.4.5.5 as indicated:*

**12.4.5.5 Construction of an SAE Confirm message**

“A peer generates a confirmation, *confirm*, and inserts it into an SAE Confirm message, by passing the KCK, the current value of the send-confirm counter (see 9.4.1.37 (Send-Confirm field)), the scalar and element from the sent SAE Commit message, and the scalar and element from the received SAE Commit message to the confirmation function CN.

**CIDs 2382, 2383, 2384, and 2385: Encoding of other components, and formats, and italicization**

CIDs 2384 and 2385: “’The peer-send-confirm shall be in the format in subclause 9.2.2 (Conventions), as extracted out of the received frame.’ -- the encoding of the other components, e.g. peer-commit-scalar, is not specified” and “’The send-confirm counter shall be in the format specified in subclause 9.2.2 (Conventions) in the order in which it is transmitted over the air.’ -- the encoding of the other components, e.g. commit-scalar, is not specified” both of which have identical proposed changes of “Add a reference to the encoding of the other components.”

CID 2382: “’The send-confirm counter shall be in the format specified in subclause 9.2.2 (Conventions) in the order in which it is transmitted over the air.’ should be Subclause, and not clear what ‘in the format means’ (encoding?)” with a proposed change of “Change to ‘The send-confirm counter is encoded in the transmitted frame per the conventions in Subclause 9.2.2.’”

CID 2383: “’The peer-send-confirm shall be in the format in subclause 9.2.2 (Conventions), as extracted out of the received frame.’ should be Subclause, not clear what ‘in the format means’ (encoding?) and wrong font for first letter in ‘peer-send-confirm’” with a proposed change of “Change to ‘The peer-send-confirm counter is decoded from the received frame per the conventions in Subclause 9.2.2.’, making sure all of ‘peer-send-confirm’ is italic [sic].”

Discussion: The other components are scalars and elements and their encoding is described in the sections that describe how the messages are encoded and decoded. The important thing that is being conveyed with “in the format of” is that it is encoded in the endianness of the frame as it goes over the air not as it was being processed by the host. The counter must be put into host order in order to properly check its value but when it is used in the computation of a verification hash it has to be in the over-the-air format.

The only occurances of a capitalized “subclause” appear when the word is the first in a sentence. And there are occurances of “subclause” followed by a section number in our standard so the word will remain uncapitalized and deference to the editor will be given to fix up this text according to the style guide (or the Style Guide as the case may be—pun intended).

Proposed resolution: Revised. See resolution in [this document].

*Instruct the editor to modify sections 12.4.5.5 and 12.4.5.6 as indicated:*

**12.4.5.5 Construction of an SAE Confirm message**

The *send-confirm*counter shall be encoded according to subclause 9.2.2 (Conventions). The elements and scalars shall be in the format they were encoded in when transmitted in an SAE Commit message as described in 12.4.7.4 (Encoding and decoding of SAE Commit messages). The message shall be transmitted to the peer as described in 12.4.7 (Framing of SAE).

**12.4.5.6 Processing of a peer’s SAE Confirm message**

The *peer-send-confirm* shall be encoded according to subclause 9.2.2 (Conventions). The elements and scalars shall be in the format they were encoded in when transmitted in an SAE Commit message as described in 12.4.7.4 (Encoding and decoding of SAE Commit messages). If the *verifier*equals *peer-confirm*, the STA shall accept the peer’s authentication and set the lifetime of the PMK to the value dot11RSNAConfigPMKLifetime. If the *verifier*differs from the peerconfirm, verification of the peer's SAE Confirm message shall fail

**No CID: Discrepency between table and state machine**

“table 9-43 does not indicate that a finite cyclic group field is sent back when status=77 but the state machine does”

Discussion: Status 77 is “Unsupported Finite Cyclic Group”. We could modify the table or modify the state machine. It does seem semantically better to indicate what you don’t like so the table should be updated.

Proposed resolution:

*Instruct the editor to modify table 9-43 as indicated:*

|  |  |  |  |
| --- | --- | --- | --- |
| SAE | 1 | Any | Scalar is present if the Status Code field is zero.  Element is present if the Status Code field is zero.  Anti-Clogging Token is present if status is 76 or if  frame is in response to a previous rejection with Status 76.  Finite Cyclic Group is present if the Status Code field is zero, 76, or 77.  Password Identifier element is optionally present if the Status Code is zero or 123 |

**No CID: Provide Guidance on Weak Diffie-Hellman groups**

“Attacks are possible against SAE when using groups defined of a prime field when that prime is less too small, when groups have a small prime order sub-group, or when they are defined over an elliptic curve with a small prime. Guidance on use of appropriate groups should be given.”

Discussion: Valid groups are touched on but only for ECC and it does not discuss prime lengths. Table 2 from SP800-57 says that 128 bits of Security Strength are obtained by FFC groups whose prime is at least 3072 bits and ECC groups whose prime is at least 256 bits. That’s what we should require because our default cipher is AES-CCM-128.

Proposed resolution:

*Instruct the editor to add the following to Section 2 as indicated:*

**2. Normative references**

NIST Special Publication 800-57, Part 1 Rev 4, Recommendation for Key Management, Elaine Barker, January 2016.

*Instruct the editor to modify section 12.4.4.1 as indicated where [x] is the reference to the normative reference above:*

**12.4.4 Finite cyclic groups**

**12.4.4.1 General**

SAE uses discrete logarithm cryptography to achieve authentication and key agreement. Each party to the exchange derives ephemeral public and private keys with respect to a particular set of domain parameters that define a finite cyclic group. Groups may be based on either Finite Field Cryptography (FFC) or on Elliptic Curve Cryptography (ECC). Each component of a group is referred to as an element . Groups are negotiated using an identifying number from a repository maintained by IANA as “Group Description” attributes for IETF RFC 2409 (IKE) [B18][B33]. The repository maps an identifying number to a complete set of domain parameters for the particular group. Not all groups defined in this repository are suitable. Only FFC groups whose prime is at least 3072 bits and ECC groups defined over a prime field whose prime is at least 256 bits are suitable for use with SAE. ECC groups defined over a characteristic 2 finite field or ECC groups with a co-factor greater than 1 shall not be used with SAE (see [x]). For the purpose of interoperability, a STA shall support group 19, an ECC group defined over a 256-bit prime order field.

**12.4.4.2.1 ECC group definition**

**References:**