|  |  |
| --- | --- |
| Project | **IEEE 802.21m Media Independent Services** **<**[**http://www.ieee802.org/21/**](http://www.ieee802.org/21/)**>** |
| Title | **Proposed remedy for Cmt i-41 in P802.21m SB** |
| DCN | **21-16-0105-00-REVP** |
| Date Submitted | **September 13, 2016** |
| Source(s) | Yoshikazu Hanatani (Toshiba) |
| Re: | Session #76, Warsaw, Poland |
| Abstract | This contribution proposes following changes for IEEE P802.21m/D04-July 2016.Add a text on a default PRF used for key derivation function of group session key. Change the definition of MULTICAST\_CAP in Table E.24 to indicate supported PRFs. |
| Purpose | Disposition detail for Cmt i-41 in P802.21m SB. |
| Notice | This document has been prepared to assist the IEEE 802.21 Working Group. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein. |
| Release | The contributor grants a free, irrevocable license to the IEEE to incorporate material contained in this contribution, and any modifications thereof, in the creation of an IEEE Standards publication; to copyright in the IEEE’s name any IEEE Standards publication even though it may include portions of this contribution; and at the IEEE’s sole discretion to permit others to reproduce in whole or in part the resulting IEEE Standards publication. The contributor also acknowledges and accepts that IEEE 802.21 may make this contribution public. |
| Patent Policy | The contributor is familiar with IEEE patent policy, as stated in [Section 6 of the IEEE-SA Standards Board bylaws](http://standards.ieee.org/guides/opman/sect6.html#6.3) <[http://standards.ieee.org/guides/bylaws/sect6-7.html#6](http://127.0.0.1:4664/cache?event_id=757737&schema_id=1&s=5X0vID10lu_E6yrIkWkNd4Wz2H8&q=hancock)> and in *Understanding Patent Issues During IEEE Standards Development* <http://standards.ieee.org/board/pat/faq.pdf> |

**Comment: i-41**

Group session key derivation in 9.6.1 support three PRF, i.e., PRF\_CMAC\_AES, PRF\_HMAC\_SHA1, PRF\_HMAC\_SHA256.
However, following items are missed.
 1. Default PRF.
 2. How to indicate supported PRFs from GM to recipients.

**Problem:**

 To derive a group session key from a master group key, all group members shall use the same PRF. In other words, interoperability is not satisfied when group members use different PRFs.

**Disposition Detail:**

1. Add a text on a default PRF used for key derivation function of group session key.

2. To indicate supported PRFs, change the definition of MULTICAST\_CAP in Table E.24 as follows. (PRF\_LIST is added.)

SEQUENCE(
LIST(UNSIGNED\_INT(1)),
LIST(UNSIGNED\_INT(1)),
PRF\_LIST,
IS\_GROUP\_MANAGER
)

1. * 1. Group key distribution Ciphersuites

The ciphersuites used for distributing a master group key are defined in Table 28.

Table —Group key distribution Ciphersuites

|  |  |  |
| --- | --- | --- |
| Code | Wrapping Algorithm | MAC AlgorithmFor VerifyGroupCode |
| 11010100 | AES\_Key\_Wrapping-128 | NULL |
| 11000100 | AES\_ECB-128 | NULL |
| 11000101 | AES\_ECB-128 | AES-CMAC-128 |
| 11000000 | No group key distribution | NULL |

In Table 28, AES\_Key\_Wrapping is an AES mode of operations specified in NIST SP 800-38F.

 Note that ECB mode is not recommended to protect a key, because it cannot provide proper security level for the key. In particular, the same plaintext is encrypted to the same ciphertext, since no random IV is used for each encryption. On the other hand, if transmitting IVs increases the size of GroupKeyData to an unacceptable point for the transport protocol, then ECB mode may be used, assuming that the same key is retransmitted with a very small probability and signature is applied to provide authentication and integrity.

The support of code “11010100” is mandatory and all entities supporting this specification shall implement it. The default PRF used for key derivation function specified in 9.6.1 is PRF\_CMAC\_AES.

Note that digital signature algorithm ECDSA-256 is used to protect group key distribution.

**Annex E.**

Table .—Data type for security

|  |  |  |
| --- | --- | --- |
| Data type | Derived from | Definition |
| ID\_TYPE | EUMERATED | The type of security association.0: TLS-generated;1: EAP-generated;2: GKB-generated |
| ID\_VALUE | OCTET\_STRING | Represents a security association identifier |
| MIS\_SEC\_CAP | SEQUENCE( TLS\_CAP, EAP\_CAP,MULTICAST\_CAP) | Represents the MIS security capabilities. |
| TLS\_CAP | BOOLEAN | TLS-generated SA capability. TRUE: (D)TLS Supported FALSE: (D)TLS not supported |
| EAP\_CAP | CHOICE( NULL, SEQUENCE(　KEY\_DIST\_LIST, 　　INT\_ALG\_LIST,　CIPH\_ALG\_LIST, 　PRF\_LIST)) | EAP-generated SA capability. When NULL is chosen, EAP-generated SA is not supported. When SEQUENCE is chosen, EAP-generated SA is supported. |
| KEY\_DIST\_LIST | BITMAP(8) | Represents a list of key distribution methods. Bitmap valuesBit 0: Push key distributionBit 1: Optimized proactive pull key distributionBit 2: Reactive pull key distributionBit 3–7 (Reserved) |
| INT\_ALG\_LIST | BITMAP(8) | Represents a list of integrity algorithm. Bitmap valuesBit 0: INTG\_HMAC\_SHA\_96Bit 1: INTG\_HMAC\_CMAC\_AES Bit 2–7 (Reserved) |
| CIPH\_ALG\_LIST | BITMAP(8) | Represents a list of encryption algorithm. Bitmap valuesBit 0: ENCR\_AES\_CBCBit 1: AUTH\_ENC\_AES\_CCM Bit 2: ENCR\_NULLBit 3-7 (reserved) |
| PRF\_LIST | BITMAP(8) | Represents a list of key derivation functions. Bitmap valuesBit 0: PRF\_AES\_CMACBit 1: PRF\_HMAC\_SHA1Bit 2: PRF\_HMAC\_SHA256Bit 3-7 (reserved) |
| NONCE\_VALUE | UNSIGNED\_INT(2) | Represents a random value |
| AUTH\_INFO\_VALUE | OCTET\_STRING | Represents the authentication information used to authenticate. |
| AUTH\_VALUE | OCTET\_STRING | Represents a message authentication/integrity code. |
| KEY | OCTET\_STRING | Represents a cryptographic key. |
| KEY\_MAPPING | LIST(SEQUENCE(　LINK\_TUPLE\_ ID, 　KEY, LIFETIME)) | Represents a map of a link layer identifier of aPoA to a key and a lifetime. |
| LL\_FRAMES | OCTET\_STRING | One or more link-layer frame(s) |
| LIFETIME | UNSIGNED\_INT(2) | Represents the period of time that a key is valid and can be used. |
| SECURITY | CHOICE(TLS\_RECORD, MIS\_SPS\_RECORD) | Represents information which is carried in the security TLV. |
| TLS\_RECORD | OCTET\_STRING | Represents a TLS record. |
| MIS\_SPS\_RECORD | SEQUENCE(ENCR\_BLOCK, CHOICE(INTG\_BLOCK,  NULL)) | Represents data protected by an MIS security association. |
| ENCR\_BLOCK | OCTET\_STRING | Represents encrypted data. |
| INTG\_BLOCK | OCTET\_STRING | Represents integrity protected data. |
| CERTIFICATE | OCTET\_STRING | Provides a X.509 Certificate |
| CERT\_BLOOM\_FILTER | SEQUENCE(OCTET\_STRING, UNSIGNED\_INT(1)) | The OCTET\_STRING part contains a Bloom Filter value computed against a set of serial numbers of revoked certificates. The UNSIGNED\_INT(1) part contains Bloom Filter parameter k. See Annex V for detailed operations. |
| CERT\_SERIAL\_NUMBER | OCTET\_STRING | Provides X.509 formatted certificate serial number, which are unique by certificate authority. |
| CERT\_SERIAL\_NUMBER\_INFO | CHOICE(LIST(CERT\_SERIAL\_NUMBER), CERT\_BLOOM\_FILTER, X509CRL) | List or Bloom Filter of X.509 certificate subfield serial numbers or X.509 Certificate Revocation List. |
| CERT\_STATUS | ENUMERATED | This indicates the status of the certificate being pushed or revoked0: Not Present – indicates that certificate is not present 1: Certificate Valid – indicates that certificate is present and that the associated public key is being used to verify signatures2: Certificate Revoked3: Certificate Expired4: Verification Failed – indicates that the signature validation of the credential failed |
| COMPLETE\_SUBTREE | CHOICE(LIST (NODE\_INDEX),COMPLETE\_SUBTREE\_BLOOM\_FILTER) | The data type for the complete subtree part of a GKB.  |
| COMPLETE\_SUBTREE\_BLOOM\_FILTER  | SEQUENCE(OCTET\_STRING, UNSIGNED\_INT(1)) | The OCTET\_STRING part contains a Bloom Filter value computed against a set of NODE\_INDEX in Complete Subtree. The UNSIGNED\_INT(1) part contains Bloom Filter parameter k.Annex P for detailed operations. |
| ENCRYPTED\_KEY\_ECB | OCTET(16) | This store a key of 16 octets encrypted with AES-ECB-128. |
| ENCRYPTED\_KEY\_KEY\_WRAP | OCTET(24) | This stores a key of 16 octets encrypted with AES-Key-Wrapping-128. |
| DEVICE\_KEY | SEQUENCE(NODE\_INDEX,NODE\_KEY) | This is the base data type of DEVICE\_KEYS. |
| DEVICE\_KEYS | LIST(DEVICE\_KEY) | The data type for a list of the device key. |
| IS\_GROUP\_MANAGER | BOOLEAN | TRUE indicates that Group Manager is supported and FALSE indicates that Group Manager is not supported. |
| GROUP\_KEY\_DATA | CHOICE(LIST(ENCRYPTED\_KEY\_ECB),LIST(ENCRYPTED\_KEY\_KEY\_WRAP),) | The data type for the key data part of a GKB. |
| GROUP\_KEY\_UPDATE\_FLAG | ENUMERATED | This indicates if the group key is to be updated.0: Key is not to be updated1: Key is to be updated |
| GROUP\_MGT\_ACTION | ENUMERATED | This indicates a manipulation command.0: Join the group1: Leave the group |
| GROUP\_STATUS | ENUMERATED | This indicates a status of group manipulation command.0: Join operation successful1: Unauthorized to join the group2: Leave operation successful3: Unchanged |
| MULTICAST\_CAP | SEQUENCE(LIST(UNSIGNED\_INT(1),LIST(UNSIGNED\_INT(1),PRF\_LIST,IS\_GROUP\_MANAGER) | The first list of UNSIGNED\_INT represents a group ciphersuite, the octet encodes a group ciphersuite code defined in Table 27. The second list of UNSIGNED\_INT represents a group key distribution ciphersuite, the octet encodes a group key distribution code defined in Table 28.  |
| NODE\_DEPTH | UNSIGNED\_INT(1) | This stores the bit length of the following NODE\_INDEX\_VALUE. |
| NODE\_INDEX | SEQUENCE( NODE\_DEPTH, NODE\_INDEX\_VALUE) | This is the base data type for COMPLETE\_SUBTREE. |
| NODE\_INDEX\_VALUE | CHOICE ( UNSIGNED\_INT(1), UNSIGNED\_INT(2), UNSIGNED\_INT(3), UNSIGNED\_INT(4) ) | This stores the index of a node of the binary tree. See 9.5 for the details. |
| NODE\_KEY | OCTET(16) | This is the base data type of a key assigned to each node in a management tree. This stores a key of 16 octets AES key. |
| RESPONSE\_FLAG | ENUMERATED | This indicates if an answer is required0: No response is needed1: Response is needed |
| SIGNATURE | SEQUENCE(CERT\_SERIAL\_NUMBER,CHOICE(SEQUENCE\_NUMBER, NULL),SIGNATURE\_DATA) | The SIGNATURE\_DATA is a digital signature data, which is verified by a verification key indicated by the CERT\_SERIAL\_NUMBER.SEQUENCE\_NUMBER shall be contained when MIS PDU protected by digital signature only. SEQUENCE\_NUMBER shall not be contained when the SIGNATURE is used as CertificateRevocation. |
| SIGNATURE\_DATA | SEQUENCE( OCTET(32), OCTET(32)) | A digital signature data. |
| SEQUENCE\_NUMBER | OCTET(10) | Sequence number used to identify a certificate. |
| SUBGROUP\_RANGE | CHOICE( SEQUENCE( UNSIGNED\_INT(1), UNSIGNED\_INT(1)),SEQUENCE( UNSIGNED\_INT(2), UNSIGNED\_INT(2)),SEQUENCE( UNSIGNED\_INT(3), UNSIGNED\_INT(3)),SEQUENCE( UNSIGNED\_INT(4), UNSIGNED\_INT(4))) | A range of valid leaf identifiers in a complete subtree of a GKB. The first integer indicates the lowest value of the range. The second integer indicates the highest value of the range. |
| SUBTREE\_FLAG | BOOLEAN | This indicates whether the leaf nodes of the complete subtree belong to the group or not. 0 (FALSE): Leaf nodes belong to the group 1 (TRUE): Leaf nodes that do not belong to the group. |
| VERIFY\_GROUP\_KEY | SEQUENCE ( OCTETS(16), OCTETS(16)) | The first OCTET(16) is arbitrary data, which is an input message to AES-CMAC (defined in IETF RFC 4493). The second OCTET(16) is the MAC value for the first OCTET(16) to be verified. |
| X509CRL | OCTET\_STRING | CertificateList defined in IETF RFC 5280. |

* + - * 1.
1.