|  |  |
| --- | --- |
| Project | **IEEE 802.15 Wireless Specialty Networks Working Group <**<http://ieee802.org/15>**>** |
| Title | **IEEE 802.15.16t Security Related Changes**  |
| Date Submitted | **2021-11-29** |
| Source(s) | Menashe Shahar (Ondas Networks) | Voice: E-mail: menashe.shahar@ondas.com |
| Re: | 16t Task Group: Licensed Narrowband Amendment |
| Abstract | Security Related Changes to IEEE 802.16-2017 |
| Purpose | Security Related Changes to 802.16-2017 with respect to the 802.16t System Requirements Document (SRD)  |
| Notice | *This document does not represent the agreed views of the IEEE 802.16 Working Group or any of its subgroups*. It represents only the views of the participants listed in the “Source(s)” field above. It is offered as a basis for discussion. It is not binding on the contributor(s), who reserve(s) the right to add, amend or withdraw material contained herein. |
| Copyright Policy | The contributor is familiar with the IEEE-SA Copyright Policy <http://standards.ieee.org/IPR/copyrightpolicy.html>. |
| Patent Policy | The contributor is familiar with the IEEE-SA Patent Policy and Procedures:<<http://standards.ieee.org/guides/bylaws/sect6-7.html#6>> and <rmation is located at <<http://standards.ieee.org/board/pat/pat-material.html>> and <<http://standards.ieee.org/board/pat>>. |

# Security related changes in 802.16-2017

The following changes to IEEE802.16-2017 are required as per the IEEE802.16t SRD.

Change

|  |  |
| --- | --- |
|  |  |
| 6.3.2.3.9Under PKM identifierPage 238 | The Identifier field is one byte. An SS uses the ID to match a BS response to the SS requests. In the case of a 3-way SA-TEK procedure, however, a BS uses it to match an SS response to the BS challenges.The SS shall increment (modulo 256) the Identifier field whenever it issues a new PKM message. ~~In PKMv1, a “new” message is an Authorization Request or Key Request that is not a retransmission being sent in response to a Timeout event. In PKMv2, a~~ PKMv2 RSA-Request, PKMv2 SA-TEK-Challenge, or PKMv2 Key-Request message is a “new” message. For retransmissions, the Identifier field shall remain unchanged...On reception of a PKM-RSP message, the SS associates the message with a particular state machine (the Authorization state machine in the case of Authorization Replies, Authorization Rejects, and Authorization Invalids for the ~~PKMv1,~~ PKMv2 RSA Reply, PKMv2 RSA Reject, PKMv2 EAP Transfer, PKMv2 SA-TEK-Challenge, PKMv2 SA-TEK-Response for the PKMv2; a particular TEK state machine in the case of Key Replies, Key Rejects, and TEK Invalids the ~~PKMv1,~~ PKMv2-Key-Reply, PKMv2-Key-Reject, PKMv2 TEK-Invalids, and PKMv2 Group-Key-Update-Command messages for the PKMv2).~~In PKMv1, an SS shall keep track of its latest ID.~~ |
| Page 240 | Auth Invalid and Auth Info messages may be used in ~~PKMv1 and~~ PKMv2. |
| 7.1.2 | The PKM protocol ~~allows for both mutual authentication and unilateral authentication (e.g., where the BS authenticates SS, but not vice versa). It also~~ supports periodic reauthentication/reauthorization and key refresh. The key management protocol uses either EAP [IETF RFC 3748] or X.509 digital certificates [IETF RFC 3280] together with RSA public-key encryption algorithm [PKCS #1] or a sequence starting with RSA authentication and followed by EAP authentication. It uses strong encryption algorithms to perform key exchanges between an SS and BS. |
| 7.1.3Page 843 | PKM supports the following two distinct authentication protocol mechanisms:— RSA protocol [PKCS #1 v2.1 with SHA-1(FIPS 186-2)] ~~(support is mandatory in PKMv1; support~~ is optional ~~in PKMv2)~~— Extensible Authentication Protocol (optional unless specifically required) |
| 7.2.2.1Page 861 | TEKs and KEKs may be either 128 bits or 256 bits long. SAs employing any ciphersuite with a basic block size of 128 bits shall use 128-bit TEKs and KEKs. Otherwise 256-bit TEKs and KEKs shall be used. The name TEK-256 is used to denote a 256-bit TEK and TEK-128 is used to denote a 128-bit TEK. Similarly,KEK-256 is used to denote a 256-bit KEK and KEK-128 is used to denote a 128-bit KEK. ~~For SAs using a ciphersuite employing DES-CBC, the TEK in the Key Reply is triple DES (3-DES) (encrypt-decrypt-encrypt or EDE mode) encrypted, using a two-key, 3-DES KEK derived from the AK.~~ For SAs using a ciphersuite employing 128 bits keys, such as AES-CCM mode, the TEK in the Key Reply isAES encrypted using a 128-bit key derived from the AK and a 128-bit block size. |
| 7.4.1.1Page 896 | ~~In PKMv1, the AK’s active lifetime a BS reports in an Authorization Reply message shall reflect, as accurately as an implementation permits, the remaining lifetimes of AK at the time the Authorization Reply message is sent. In PKMv2,~~ AK lifetime is determined by either PMK lifetime or PAK lifetime, or both ofthem.  |
| 7.4.1.2Page 896Remove paragraphs 2,3 | ~~In PKMv1, an AK transition period begins when the BS receives an Auth Request message from an SS and the BS has a single active AK for that SS. In response to this Auth Request, the BS activates a second AK [see point (a) and (d) in Figure 7-14], which shall have a key sequence number one greater (modulo 16) than that of the existing AK and shall be sent back to the requesting SS in an Auth Reply message. The BS shall set the active lifetime of this second AK to be the remaining lifetime of the first AK [between points (a) and (c) in Figure 7-14], plus the predefined~~ *~~AK Lifetime~~*~~; thus, the second, “newer” key shall remain active for one~~ *~~AK Lifetime~~* ~~beyond the expiration of the first, “older” key. The key transition period shall end with the expiration of the older key. This is depicted on the right-hand side of Figure 7-14.~~~~As long as the BS is in the midst of an SS’s AK transition period, and thus is holding two active AKs for that SS, it shall respond to Auth Request messages with the newer of the two active keys. Once the older key expires, an Auth Request shall trigger the activation of a new AK, and the start of a new key transition period.~~ |
| 7.4.1.3Page 897 | A BS shall use a HMAC/CMAC\_KEY\_U (see 7.5.4.3 and 7.5.4.4) derived from one of the SS’s active Aks to verify the CMAC/HMAC Digest in ~~Key Request/~~PKMv2-Key-Request messages received from the SS. The AK Key Sequence Number accompanying each Key Request/PKMv2-Key-Request message allows theBS to determine which HMAC/CMAC\_KEY\_U was used to authenticate the message. ~~In PKMv1, if the AK Key Sequence Number indicates the newer of the two AKs, the BS shall identify this as an~~ *~~implicit acknowledgment~~* ~~that the SS has obtained the newer of the SS’s two active AKs [see points (b) in Figure 7-14]~~.A BS shall use a HMAC/CMAC\_KEY\_D derived from the active AK selected above (see also 7.5.4.3 and 7.5.4.4) when calculating CMAC/HMAC Digests in ~~Key Reply/~~PKMv2-Key-Reply, ~~Key Reject/~~PKMv2-Key-Reject, and ~~TEK Invalid/~~PKMv2-TEK-Invalid messages. When sending ~~Key Reply/~~PKMv2-Key-Reply, ~~Key Reject/~~PKMv2-Key-Reject, or ~~TEK Invalid/~~PKMv2-TEK-Invalid messages within a key transition period (i.e., when two active AKs are available), if the newer key has been implicitly acknowledged, the BS shall use the newer of the two active AKs. If the newer key has not been implicitly acknowledged, the BS shall use the older of the two active AKs to derive the KEK and the HMAC/CMAC\_KEY\_D. |
| Page 898 | A BS shall use a HMAC/CMAC\_KEY\_U (see 7.5.4.3 and 7.5.4.4) derived from one of the SS’s active Aks to verify the CMAC/HMAC Digest in ~~Key Request/~~PKMv2-Key-Request messages received from the SS. The AK Key Sequence Number accompanying each ~~Key Request/~~PKMv2-Key-Request message allows theBS to determine which HMAC/CMAC\_KEY\_U was used to authenticate the message. ~~In PKMv1, if the AK Key Sequence Number indicates the newer of the two AKs, the BS shall identify this as an~~ *~~implicit acknowledgment~~* ~~that the SS has obtained the newer of the SS’s two active AKs [see points (b) in Figure 7-14].~~The BS shall use a KEK derived from an active AK when encrypting the TEKs in the ~~Key Reply/~~PKMv2-Key-Reply messages. ~~The right-hand side of Figure 7-14 illustrates the BS’s policy regarding its use of Aks in PKMv1, where the shaded portion of an AK’s lifetime indicates the time period during which that AK~~~~shall be used to derive the HMAC/CMAC\_KEY\_U, HMAC/CMAC\_KEY\_D, and KEK.~~ |
| 7.4.2Page 899 | **7.4.2 SS key usage**In ~~PKMv1 or~~ PKMv2 RSA-based authentication, the SS is responsible for sustaining authorization with its BS and maintaining an active AK. In PKMv2 EAP-based authentication, reauthorization can be initiated by either BS or SS to refresh the AK. An SS shall be prepared to use its two most recently obtained AKsaccording to the manner described in 7.4.2.1 through 7.4.2.3.**7.4.2.1 SS reauthorization**AKs have a limited lifetime and shall be periodically refreshed. ~~In PKMv1, an SS refreshes its AK by reissuing an Auth Request to the BS. The Authorization state machine (7.2.1.5) manages the scheduling of Auth Requests for refreshing AKs.~~ In PKMv2 RSA-based authentication, the SS refreshes its AK by issuing a PKMv2 RSA-Request message. In PKMv2 EAP-based authentication, reauthorization can be initiated by either BS or SS to refresh the AK. The SS initiates reauthorization by issuing PKMv2 EAP-Start message to the BS. The BS initiates reauthorization by issuing PKMv2 EAP-Transfer message encapsulating EAP request/identity to the SS. The authorization state machine for PKMv2 EAP-based authentication is described in 7.2.2.8 |
| Page 900 | ~~In PKMv1, an SS’s Authorization state machine schedules the beginning of reauthorization a configurable duration of time, the~~ *~~Authorization Grace Time~~*~~, [see points (x) and (y) in Figure 7-14], before the SS’s latest AK is scheduled to expire. The Authorization Grace Time is configured to provide an SS with an authorization retry period that is sufficiently long to allow for system delays and provide adequate time for the SS to successfully complete an Authorization exchange before the expiration of its most current AK.~~ |
| 7.4.2.2Page 901 | The SS shall be able to use the HMAC/CMAC\_KEY\_D derived from either of its two most recent AKs to authenticate Key Reply, Key Reject, and TEK Invalid messages for ~~PKMv1, or~~ PKMv2-Key-Reply, PKMv2-Key-Reject, and PKMv2-TEK-Invalid messages for PKMv2. The SS shall be able to decrypt an encrypted TEK in a ~~Key Reply or~~ PKMv2-Key-Reply message with the KEK derived from either of its two most recent AKs. The SS shall use the accompanying AK Key Sequence Number to determine which set of keying material to use.~~The left-hand side of Figure 7-14 illustrates an SS’s maintenance and usage of its AKs in PKMv1, where the shaded portion of an AK’s lifetime indicates the time period during which that AK shall be used to decrypt TEKs. Even though it is not part of the message exchange, Figure 7-14 also shows the implicit acknowledgment of the reception of a new AK via the transmission of a Key Request message using the key~~~~sequence of the new AK.~~ |
| 7.5.9Page 917 | As with its RSA encryption keys, Privacy uses 65537 (0x010001) as the public exponent for its signing operation. Manufacturer CAs shall employ signature key modulus lengths of at least 2048 bits ~~and no greater than 2048 bits~~. |
| Page 1878 to  | **13.1.3.5.3 wmanIf2BsPkmV1Objects** |
| 11,9.2Page 1683 | This attribute contains a 128 byte quantity containing the AK RSA-encrypted with the SS’s 4096 bit RSA public key. Details of the RSA encryption procedure are given in 7.5. The ciphertext produced by the RSA algorithm shall be the length of the RSA modulus, i.e., 128 bytes |
| Page 1689 | Add: 4 CCM mode 256-bit AES5 CBC mode 256-bit AES |
|  | Add:2 CCM mode. 256-bit AES |
|  | Add:5 ECB mode AES wit 256-bit key6 AES key wrap with 256-bit key |
| 1690 | Add:

|  |  |
| --- | --- |
| 0x000005 | No data encryption, no data authentication and RSA 2048 |
| 0x000006 | No data encryption, no data authentication and RSA 4096 |
| 0x040007 | CCM mode AES 256-bit, no data authentication and AES, 256 |
| 0x040207 | CCM mode AES 256-bit, CCM mode 256-bit, ECB mode AES with 256-bit key |
| 0x040208 | CCM mode AES 256-bit, CCM mode 256-bit, AES key wrap with 256-bit key |
| 0x050007 | CBC mode AES 256-bit, no data authentication, ECB mode AES with 256-bit key |

 |
| 13.1.3.5.3 | Remove |