# IEEE P802.15
# Wireless Personal Area Networks

| | | | |
|---|---|---|---|
| Project | IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs) | | |
| Title | **IEEE P802-15_TG3 D10 Security Related Comment Resolutions** | | |
| Date Submitted | [July 23, 2002] | | |
| Source | [Ari Singer, Daniel V. Bailey]<br>[NTRU]<br>[5 Burlington Woods<br>Burlington, MA 01803 USA] | Voice:<br>Fax:<br>E-mail: | [+1 781 418-2515]<br>[+1 781 418-2532]<br>[asinger@ntru.com] |
| Re: | 802.15.3 TG3 Letter Ballot Draft D10 | | |
| Abstract | [This document is offered as rolling recommended resolutions for security related ballot comments on 802.15.3 D10.] | | |
| Purpose | [This document is offered as rolling recommended resolutions for security related ballot comments on 802.15.3 D10. It will be updated frequently to accommodate input and decisions by the working group as well as adding more proposed resolutions for other ballot comments.] | | |
| Notice | This document has been prepared to assist the IEEE P802.15. 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 acknowledges and accepts that this contribution becomes the property of IEEE and may be made publicly available by P802.15. | | |

# 1. Comment resolution, Vancouver to Schaumburg

## 1.1 Week of July 22, 2002

### 1.1.1 SEC (General)

433 (Gilb, T) The SECID, time token and integrity code fields are not defined before they are first discussed. Add either a forward reference to the definitions of these fields or define them here or in 7.2 with a generic secure frame as an example. **Suggest accept in principle. Recommend replacing figure 6 with the following figure.**

| 0-4 | 0-8 | variable | 0-6 | 0-2 | 2 | 1 | 3 | 1 | 1 | 2 | Octets: 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FCS | Integrity code | Frame body | Time token | SECID | HCS | Stream Index | Frag. Control | Source DEVID | Dest. DEVID | PNID | Frame Control |
| MAC frame | | | | | | MAC header | | | | | |

**Figure 1—MAC header and frame format**

**Recommend changing the value for the maximum frame length in 7.2.7 to aMaxFrameSize-20.**

**Recommend adding the following sub-clauses to 7.2:**

**SECID field**

The SECID field contains a 2-octet identifier for the key that is being used to protect the frame. The SECID for a given key is selected by the security manager in the secure relationship as described in {xref - see resolution to 224 and 846}. The SECID for management keys is communicated to a DEV in a successful authentication protocol by the security manager in the challenge request command {xref - 7.5.2.3}. The SECID for data keys is communicated to a DEV by the security manager in a distribute key request command {and broadcast distribute key command pending resolution to Odman's e-mail}, 7.5.2.7, or a request key response command, 7.5.2.6.

If the SEC bit in the frame control field is set to 0, the SECID shall not be sent.

**Time token field**

The time token field contains a 6-octet {pending resolution to 776} counter that is incremented each time a beacon is transmitted. The time token is used to provide a unique sequence number for the beacon and to provide freshness on secure frames transmitted within that superframe.

The time token field shall be sent in all secure beacon frames and may be sent in insecure beacon frames. The time token field shall not be sent in non-beacon frames.

**Integrity code field**

The integrity code field contains an 8-octet encrypted integrity code that is used to cryptographically protect the integrity of the MAC header and MAC frame. The integrity code is computed as specified in {xref - 10.2.5}.

If the SEC bit in the frame control field is set to 0, the integrity code field shall not be sent.

862 (Shvodian, T) Initial Owner needs a definition. Definine initial owner. **Suggest accept. Recommend the following text be inserted into sub-clause 9.9:**

For each protocol described in this sub-clause, tables are included to specify the requirements for the DEV and security manager to successfully implement the protocol. The setup table specifies the required data that must be stored by each device, denoted the initial owner, before the protocol is initiated. The capabilities table specifies the required functionality for each device to perform its respective role in the protocol.

930 (Shvodian, T) Need to make sure that all fields specified as ( a || b || c) are msb to the left, first bit transmitted to the right. Make sure this is consistent with the rest of the draft. **Resolve along with comment 150.**

### 1.1.2 SECID

870 (Shvodian, TR) What does a DEV do when it sees a new SECID in the beacon? Does it stop tranmitting? If it keeps transmitting with the old SECID can/shall another DEV use the old key? Need to clarify. **Suggest accept in principle. Resolve along with comment 941.**

941 (Shvodian, T) It may be necessary to allow a SECID to be used for n frames after the SECID has been updated incase a DEV did now see the SECID change in the beacon. DEVs with pseudo-static slots are able to transmit even if some number of beacons are corrupted. Decide if this should be allowed an update the text. **Suggest accept in principle. Recommend adding the following text to clause 9.3 (related to text in 4.4 of 02/273r5):**

**Changes in the piconet-wide group data key**

When the PNC changes the piconet-wide group data key, the PNC shall transmit the new key to all of the currently authenticated DEVs using the {xref - distribute key command or broadcast distribute key command pending resolution of Odman’s e-mail comment}. When a DEV receives a valid {xref - distribute key or broadcast distribute key} command from the PNC, the DEV shall use the new key for all outgoing secure frames that require the use of the piconet-wide group data key. The DEV may continue to accept frames protected by the old piconet-wide group data key for up to {65,535 ms}.

If a DEV that is in the AWAKE state, {xref - 8.12}, receives a beacon with a time token greater than the last known time token, but with a SECID that does not match the SECID of the known key, the device shall send a key request command to the PNC to obtain the new key. While waiting to obtain the new key, the DEV may accept the new time token value and continue to transmit and accept frames with the last known piconet-wide group data key for up to {65, 535 ms}.

A DEV, upon entering the AWAKE state from the SLEEP state, {xref - 8.12}, shall not transmit or accept frames protected with the piconet-wide group data key until it receives a valid beacon protected with the known key. If it instead receives a beacon with a SECID that is not the same as the SECID corresponding to the last known piconet-wide group data key, the DEV shall securely delete the stored piconet-wide group data key and send a request key command, {xref - 7.5.2.6}, to the PNC to obtain the new key.

224 (Gilb, TR) Because the SECID is now a 2-byte value, there is a reasonably high probability that multiple keys will share the same SECID. Since there is only one SECID currently in use for a particular relationship and type of key (management key or data key), there should be an indication in the SECID about what kind of key it is to avoid collisions that will cause confusion. Use the msbs of the SECID to differentiate the type of keys for which it is associated. Add text where the SECID is defined that says that "The msb of the SECID shall be set to one for PNC-DEV keys and shall be set to 0 for peer-to-peer keys. The next most significant bit shall be set to 1 for data keys and shall be set to 0 for management keys." A table may work better. **Resolve along with comment 846.**

846 (Shvodian, TR) SECIDs can't be unique in the piconet unless they are either centrally managed or the SM is identified with each frame. How does a dev know if the key is the group key assigned by the PNC or a group key assigned by another SM. Also, the receiver needs to decode command frame types to know which key is used for commands. Security Editors need to come up with a way to let the recreiving DEV identify which key is being used. **Accept in principle. Recommend the following text be added to clause 9.3 along with the text specified in 4.4.1 of 02/273r5:**

**Selecting the SECID for new keys**

For each management and data key used in the piconet, the security manager in the relationship shall select the 2-octet SECID that identifies the key. The first octet of the SECID for all keys except the piconet-wide group data key shall be set to the DEVID of the security manager in the relationship. The second octet shall be set to the value of a 1-octet roll-over counter that is used to number the keys shared with the DEV in that security relationship. The SECID for the piconet-wide group data key shall have the first octet set to the BcstID, {xref - 7.2.3}, and the second octet shall be set to the value of a 1-octet roll-over counter that is used to number the piconet-wide group data keys.

565 (Gilb, TR) 7.5.2.6-7.5.2.9: The security session ID (SECID) should be included before the Encrypted Seed (where the sequence number currently resides) in the request key response, distribute key request and distribute key response commands. This value is needed to uniquely identify the key that is being transmitted in the protocol. Note that the SECID should not be included in the request key command since the requesting party may not know the SECID of the key being requested. Delete the SECID from the key request command. Change the name of the SECID field in the other three commands to be Key SECID. Add the following text to each of the three commands: The key SECID field is the unique identifier for the seed (and corresponding key) that is being transported in this protocol. **Suggest accept.**

### 1.1.3 Secure ACK

843 (Shvodian, T) Add the ACKs to the figures unless it makes them unnecessarily complicated. Otherwise, leave it as is. Change from integrity protected ACK to Immediate ACK. **Suggest accept.**

927 (Shvodian, T) Secure ACK is not needed. Remove the Secure ACK message authentication generation. **Suggest accept.**

282 (Shvodian, TR) Remove Secure Immediate ACK. It serves no purpose and complicates the ACK frame by giving it a frame body. Delete Secure Imm-ACK frame. **Suggest accept.**

457 (Gilb, TR) There are no ACKs shown in the overview figures. Add the ACKs to the figures unless it makes them unnecessarily complicated. Otherwise, leave it as is. **Suggest reject. Since the ACK mechanism is simply used to help ensure a more reliable communications medium, it does not seem to relate directly to the protocol overviews. However, if this would improve clarity, the ACKs may be added with minimal clutter in the diagrams.**

### 1.1.4 ACL

852 (Shvodian, T) Does SM check ACL after getting association request? Need a figure showing SM checking ACL after association. **Suggest accept in principle. The association request is only sent to the PNC. When in modes 1, 2 or 3, the PNC may choose to not allow a device to remain in the piconet based on the ACL if desired, but this should occur based on the authentication protocol. Recommend adding a NULL security suite that shall be used in mode 1 (no cryptographic operations are performed in this security suite, only an ACL check). Recommend adding additional text explaining that a PNC may choose to disassociate a device that fails the authentication.**

221 (Gilb, T) Each entry in the access control should be able to support keys shared with that particular device. For each access control list table, there should be ManagementKeyInfo, ManagementSECID, DataSECID, DataKeyInfo entries. Adding these fields to the table. **Suggest accept in principle. The DEV needs to possess management and data keys for each relationship. If the PIB remains in a similar form, these entries should be added.**

### 1.1.5 Beacon

776 (Shvodian, TR) It is a waste to have a 6 octet time token in a secure beacon and a 4 octet beacon number in the piconet synchronization parameter. Are 6 octets really needed? Octets would roll over less than once per year with a 10 ms superframe. If 4 octets are sufficient, just use the beacon number. If 6 octets are needed, change the beacon number in the piconet synchronization parameter to 6 octets and delete the time token. **Suggest accept in principle. Recommend using a 6-byte time token and remove the beacon number (or call the 6-byte thing the beacon number). Some devices may end up choosing a starting beacon number that is not zero and if superframe length decreases, it seems preferable to not have to deal with rollover (which forces rekeying) when possible. This is also less of an issue if the time token is not included in each of the frames.**

387 (Heberling, TR) Insert a copy of table 38 into clause 7.3.1.2 just before Table 40 with these info elements for the secure beacon frame . . . **Suggest accept.**

### 1.1.6 Auth

936 (Shvodian, T) An authenticated DEV can use the probe command. Can an unassociated DEV? If the PNC is checking the ACL to determine association privliges, a DEV could get refused from associating. Clarify if an associated DEV can do a probe. Split unauthenticated into two columns: unassociated and associated. **Suggest accept in principle. Text will be updated to clarify an associated but unauthenticated DEV may send probe commands. Unassociated DEVs shall send only association request commands. An associated but not authenticated DEV may send a probe command.**

931 (Shvodian, TR) This raises an interesting question: "If the hash is not in the PIB, the public key is passed to the DME to establish trust by other means." Is the security function in the DME? The MLME_request.indication goes up to the SM's DME. So is the SM part of the DME? Need to clarify where the security function resides in the reference model of figure 3. Is it part of the DME? **Suggest accept in principle. The security manager operations, which consist of managing the keys for the relationship, reside in the DME. The DME also maintains the ACL, which is used for managing the keys.**

864 (Shvodian, T) All of these states need to specify that the DEV ignores Beacon integrity. **Suggest accept in principle.**

310 (Shvodian, T) Authentication response command needs a response value of "DEV not a security manager" in case a DEV tries to associate with another DEV who is not a security manager. Add a "DEV is not a security manager" response code. **Suggest accept.**

856 (Shvodian, T) Add association to the list of commands that the SM handles in startup state. **Suggest accept.**

## 1.2 Broadcast Distribute Key Command

A comment was made (is this a ballot comment?) that the distribute key commands may get to be too costly if they need to be sent individually to each device whenever a rekey is to take place. Here is some proposed text for a single broadcast distribute key command. There are also several other places that may require updates if this command is added.

**Note: The command is defined with the use of AES-CCM with an 8-byte integrity code.**

### 1.2.0.1 Broadcast distribute key command

The broadcast distribute key command is used by the PNC in a distribute key (“push”) protocol to transmit a new group piconet data key to all of the authenticated DEVs in the piconet.

The ACK request shall be set to No-ACK . The SEC field shall be set to 0. The frame control Dly-ACK policy sub-field in the MAC header of this command shall be set to zero and shall be ignored upon reception.

The broadcast distribute key command shall be formatted as illustrated in Figure 2.

| 28 | 1 | ... | 28 | 1 | 2 | 2 | 2 |
|---|---|---|---|---|---|---|---|
| Encrypted seed block n | DEVID | ... | Encrypted seed block 1 | DEVID | Seed SECID | Length (=2+n*29) | Command type |

**Figure 2—Broadcast distribute key command format**

The seed SECID shall be the SECID of the seed that is encrypted in each of the encrypted seed blocks.

The PNC shall include one encrypted seed block for each authenticated DEV in the piconet. The encrypted seed block shall be formatted as illustrated in Figure 3.

| octets: 8 | 16 | 2 | 2 |
|---|---|---|---|
| Integrity code | Encrypted seed | Secure frame counter | SECID |

**Figure 3—Encrypted seed block format**

The SECID shall be the SECID of the management key shared between the PNC and the DEV specified by the preceding DEVID.

The secure frame counter is the unique secure frame counter used by the PNC for secure frames in the current superframe.

The encrypted seed is as defined in the security suite, Clause 10.

The integrity code provides integrity on the encrypted seed block and is generated as specified in the security suite, Clause 10.

### 1.2.0.2 Symmetric key operations (clause 10.2.5.2)

Add the following entries to table 82:

**Table 1—Symmetric cryptographic operations**

| Operation | Specification |
|---|---|

**Table 1—Symmetric cryptographic operations**

| | |
|---|---|
| Encrypted seed block integrity code | The integrity codes included in the encrypted seed blocks in broadcast distribute key command frames are generated by computing the encrypted integrity code using CCM authentication and encryption as specified in 10.2.4.3. This operation shall be performed using the management key specified by the SECID in the encrypted seed block. The DEVID for the encrypted seed block shall be used as the destination DEVID for the nonce, the secure frame counter in the encrypted seed block shall be used as the secure frame counter for the nonce, the authentication data input *a* shall be the 2-byte seed SECID and the 16-byte pre-encrypted seed shall be the plaintext input *m* for encryption (and authentication). |
| Encrypted seed block seed encryption operation | The seed for key transport is encrypted using CCM authentication and encryption on the seed as specified in 10.2.4.3 This operation shall be performed using the management key specified by the SECID in the encrypted seed block. The DEVID for the encrypted seed block shall be used as the destination DEVID for the nonce, the secure frame counter in the encrypted seed block shall be used as the secure frame counter for the nonce, the authentication data input *a* shall be the 2-byte seed SECID and the 16-byte pre-encrypted seed shall be the plaintext input *m* for encryption (and authentication). |

## 2. Resolved Comments

460 (Gilb, T) There is no introductory text to describe this subclause. Text is also missing from 9.9.4 and 9.9.6. **Accept in principle.**

Text for clause 9.9.3 introduction: In a secure piconet or in a secure peer-to-peer relationship, the security manager may wish to update the current data protection key by initiating the distribute key protocol described here. For a change in the piconet group data key, the PNC sends the new piconet group data key to each authenticated DEV before changing the key using the distribute key protocol. For a change in a peer data key, the security manager in the relationship initiates the distribute key protocol.

Text for clause 9.9.4 introduction: In a secure piconet, if a DEV receives a frame or beacon with an unknown SECID, it may initiate the request key protocol described here in order to obtain the unknown key from the security manager of the relationship.

Text for clause 9.9.6 introduction: When a DEV transmits (or recieve) a secure data frame, the DEV shall protect (or verify) the frame using the data protection protocol described here.

630 (Gilb, T) The word "can" is use when it should be "may". **Accept.** Change "The only state a DEV can " to "The only state a DEV may "

482 (Gilb, TR) The PNCs DEV address is no longer in the beacon. Ensure that the DEV address of the PNC is available in some other manner to all DEVs to peform the required security processes. **Accept in principle.** Add text in Figure 149 "Store ID_SM as the DEV address of the SM for this authentication." On page 133, line 9 change "requesting DEV" to "security manager".

426 (Gilb, T) Missing definitions for the following acronyms: CCM, DER, ECQV, ECIES, CTR, CBC, CRL, SECID. Add the following definitions: CCM - counter-counter mode, DER - ?, ECQV - eliptic curve Qu-Vanstone, ECIES - eliptic curve ??, CTR - counter mode, CBC - ??, CRL - ??, SECID - security identifier. **Accept in principle.** CCM - CTR encryption + CBC-MAC, DER - Distinguished Encoding Rules, ECMQV - Elliptic Curve Menezes-Qu-Vanstone key establishment protocol, CTR - Counter mode, CBC -

Cipher Block Chaining, CRL = Certificate Revocation List, SECID - Security Identifier, CBC-MAC = Cipher Block Chaining-Message Authentication Code

578 (Gilb, T) The comparison with TLS needs to be modified to indicate the use of CCM rather than HMAC with SHA-256 and CBC encryption. Change the comment after the first bullet to: The security suite specfication in this document specified the use of AES in CCM mode, which provides an AES CBC-MAC encrypted using AES CTR encryption. **Accept in principle.** Change "The security suite specification in this document specifies the use of HMAC with SHA-256." to "The security suite specifications in this document may specify other algorithms."

475 (Gilb, TR) Step 4 says to validate the content of ICU but does not specify how it is done. Provide the figure that was intended here and fix the xref. Otherwise, delete the sentence. **Accept in principle. This mechanism should be specified in the security suite, not in the general scheme. Struik will provide update to the implicit certificate methods and implicit certificate security sub-suite.**

474 (Gilb, TR) Figure 12 is not in the annex nor is it a valid cross reference. Specify how this validation is to be performed. Otherwise, delete the implicit certificate scheme. **Accept in principle. Struik will provide update to the implicit certificate methods and implicit certificate security sub-suite.**

293 (Shvodian, TR) & 304 (Shvodian, TR) **Reject. These IEs are used for the probe command.**

494 (Gilb, TR) The SECID is listed as an octet string in some of these tables. Change the SECID to be 2 octets in all locations. Particularly, change tables 11, 12, 13 and 32. **Accept.**