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 22, 2002]		
Source	[Ari Singer, Daniel V. Bailey] [NTRU] [5 Burlington Woods Burlington, MA 01803 USA]	Voice: Fax: E-mail:	[+1 781 418-2515] [+1 781 418-2532] [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)

862 (Shvodian, T) Initial Owner needs a definition. Definine initial owner. **Suggest accept.**

460 (Gilb, T) There is no introductory text to describe this subclause. Text is also missing from 9.9.4 and 9.9.6. **Suggest accept in principle. The distribute key protocol may be modified as per Odman's e-mail and comment 868 if the group wishes to avoid a large number of distribute key commands, but in any case, an introduction should be included.**

Proposed 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.

Proposed 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.

Proposed 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". **Suggest accept.**

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. **Suggest accept in principle. The PNCs DEV address is in the association response command and in the challenge request command. Recommend that we mention that the DEV may also request the PNC DEV address in a probe command before authentication as well.**

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. **Suggest reject. Most cryptographic specifications are written with the first byte to the left and the most significant bit (within a byte) to the left. The referenced cryptographic specifications are written in this manner, as are the CCM and implicit certificate specifications in the appendix.**

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. **Suggest accept in principle. CCM = CTR encryption + CBC-MAC, CBC = Cipher Block Chaining, CBC-MAC = Cipher Block Chaining-Message Authentication Code, CRL = Certificate Revocation List, ECIES = Elliptic Curve Integrated Encryption Scheme, DER = Distinguished Encoding Rules**

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. **Suggest accept.**

1.1.2 SEC - Implicit Certificate Specification

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. **Suggest 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. **Suggest accept in principle. Struik will provide update to the implicit certificate methods and implicit certificate security sub-suite.**

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 1.

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 1—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 2.

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

Figure 2—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
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).

Table 1—Symmetric cryptographic operations

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).