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 25, 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)

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**	**0-2**	**0-6**	**0-2**	**2**	**1**	**3**	**1**	**1**	**2**	**Octets: 2**
FCS	Integrity code	Payload	SFC	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-22.

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.

Secure frame counter (SFC) field

The secure frame counter field contains a 2-octet counter that is used to ensure the uniqueness of the nonce in a secure frame. A DEV shall not reuse a frame counter with the same time token and key. The DEV may initialize the secure frame counter at 0 and increment it each time a secure frame is sent. When the time token is updated, the DEV may reset the secure frame counter to 0 if desired or allow the counter to roll over.

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

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.

1.1.2 SECID

865 (Shvodian, TR) It needs to be clear which commands use secure command format and which use non secure command format. **Suggest accept in principle. Recommend adding table and text from 4.4.1 of 02/273r5.**

577 (Gilb, TR) There needs to be an explanation of what keys are used with which commands. clause 9 seems like a good place to put this. A table needs to be added to list the usage of the frames and the types of keys used for each frame (the table is in document 02/271r0). The following text should be added at the end of the clause describing secure frame generation: The key used to protect a particular frame depends on the purpose of the frame. In general, all secure commands between the PNC and other devices should be protected with the PNC management key. All secure data frames to or from the PNC, all secure broadcast frames and all secure beacons should be protected with the piconet group data key. For two DEVs that share a peer-to-peer security relationship, peer-to-peer management keys should be used for all secure commands and peer-to-peer data keys should be used for all secure data frames. If two DEVs in a secure piconet do not have a peer-to-peer security relationship, they may use the piconet group data key for secure commands and secure data frames transmitted between them. The following table summarizes which keys should be used for each type of frame. **Suggest accept in principle. Resolve along with 865.**

218 (Gilb, TR) The use of the SECID in the MLME-REQUEST-KEY.request and MLME-REQUEST-KEY.indication implies that the requesting device knows the SECID of the key it is requesting. This will be true for piconet-wide keys because the SECID will be included in the beacon, but for peer-to-peer keys, the DEV may not know the SECID of the current key, in which case it perhaps should be allowed to request the key without knowing its SECID. Change the MLMEs to indicate that a DEV is able to send the request key without knowing the SECID of the current key. Otherwise, perhaps the SECID can be deleted from the request command? **Suggest accept in principle. Resolve along with 565 (already resolved).**

1.1.3 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.**

1.1.4 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. If this is done, recommend keeping the time token outside of the piconet synchronization parameters.**

780 (Shvodian, TR) Remove the Time token. This can be replaced by the beacon counter in the piconet synchronization IE. **Suggest accept in principle. Recommend instead removing the beacon counter from the piconet synchronization IE and requiring that the time token be included in all beacon 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.**

569 (Gilb, TR) Need to add a description on how to create and receive a secure beacon. Add the following text to the end of subclause 9.3 9.3.6 Secure beacon processing 9.3.6.1 Generating secure beacons A PNC in a piconet using security should send secure beacons protected with the piconet protection key stored. For each superframe, the PNC should increment the time token and transmit a secure beacon with the SEC field in the frame control field set to 1. 9.3.6.2 Receiving secure beacons In order to maintain secure and reliable operations in the piconet, a DEV shall use the beacon to help maintain the current time token and the current key. When the DEV receives a secure beacon, it shall verify that the time token is greater than the current time token, that the SECID matches the SECID for the piconet and that the integrity code passes. If all of these checks succeed, the DEV shall set the current time token to be the received time token value. If the time token is greater than the current time token, but the SECID does not match the current SECID, the device may set the current time token to the value in the beacon and send a key request command to the PNC to obtain the new key. **Suggest accept in principle. Recommend merging this text with the text proposed for the resolution to 870 and 941.**

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: Info Elements Present in beacon ChannelTimeAllocation In every beacon Piconet BSID In every beacon DevAssociation As needed StreamAnnouncement As needed PNCHandoverCount As needed Piconet parm change As needed Parent PNC DEV Address As needed Integrity code In every beacon. **Suggest accept.**

296 (Shvodian, TR) Key number is no longer needed. This was added to let a DEV know when the group key changed. Since the SECID is in every beacon, DEVs will know when the key changes. Remove Key number for the beacon in figure 23 and remove the description from the text. **Suggest accept.**

1.1.5 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.**

805 (Shvodian, TR) There are many places in the draft that refer to things that an associated DEV can do. Unfortunately, with security turned on, many of these really require authentication. One solution would be to to say "associated or authenticated if required". the preferred way would be to have DEVs in mode 0 and 1 automatically authenticated in modes 0 and 1. Add text that associated DEVs are automatically authenticated in modes 0 and 1, and throughout the draft use authenticated instead of associated as appropriate. **Suggest accept in principle. The idea was proposed to include a NULL security suite that indicates to each that they agree to use no key for the piconet. This could be a 2-pass protocol using the authentication request and authenticate response commands with a NULL public key and a 0 integrity code (or something). Should we do this? Then you “authenticate” in mode 1 and mode 0 as well.**

1.1.6 De-authenticate

59 (Heberling, TR) Deauthentication cannot "fail". Both PNC and client shall regard a deauthenticate request as being completed when requested and proceed with the deauthentication procedure. The PNC needs to get back the DevID from the confirm in case it has deauthenticated several DEVs. The reasonCode is not needed since the request cannot fail, and even if it did there is no recovery./KO MLME_DEAUTHENTICATE.confirm <change text in line 7> This primitive reports the completion of a deauthentication. <Change parameter to MLME_DEAUTHENTICATE.confirm> MLME_DEAUTHENTICATE.confirm (DevID) <Change text in 6.3.10.3.1> This primitive is sent by the MLME after sending a deauthentication request command to a DEV and completeing the deauthentication procedure. The primitive shall be sent even if the deauthenticated DEV does not ACK the command frame. **Suggest accept.**

334 (Heberling, TR) What is a deauthentication acknowledgement? /KO Replace with Imm-ACK, unless a real frame is intended but missing in the frame formats. In that case insert that frame into clause 7. **Suggest accept in principle. Replace with Imm-ACK.**

1.1.7 CCM

497 (Gilb, TR) Add the text required to implement 2 key CCM, indicating that it is an option. That way, if an attack is found, the standardized implementation is already written and implementers simply need to switch over to it. **Suggest reject. The change to 2-key CCM is non-trivial and may cause some confusion.**

888 (Shvodian, TR) Is the secure frame counter 2 octets or 4? It looks like it is currently 2 octets in the data frames and 4 in the command frames. If this is the case, then a separate nonce is needed for command frames. Clarify the number of octets in the data, command and beacon secure frame counters. **Suggest accept in principle. The secure frame counter is always 2 octets. The sequence counter in commands was created for a different purpose, but should be removed. Recommend removing the 4-byte sequence counter and inserting the 2-byte secure frame counter into commands and inserting the 2-byte secure frame counter in the other secure frames.**

495 (Gilb, TR) Add a field, secure frame counter, to every secure frame. Make it 2 octets long. Add a new element called the "secure frame counter." to every secure frame. The secure frame counter basically counts the number of secure frames that a particular DEV has transmitted within that superframe. The secure frame counter shall have a length 2-bytes and go directly after the time token. This counter is used as an input to the nonce for payload protection. Add the requirement that a DEV shall not send two secure frames within the same superframe with the same secure frame counter. The simplest way to ensure this is that the beginning of each superframe, the value shall be set to 0 and it shall be incremented each time it is used within that superframe (which is any time you send a secure frame). **Suggest accept in principle.**

891 (Shvodian, T) Can one secure frame counter be used for all transmission or is a separate on needed for all groups? Clarify if it is acceptible for one secure frame counter to be used for alll frames. **Suggest accept in principle. Recommend adding secure frame counter to the general frame format in clause 7.2 as proposed for resolution to 433 including text describing how the secure frame counter is to be used. The answer to the question is that the secure frame counter may be used for all transmissions as long as no more than 2^16 frames are sent using the same time token. If more than that may be sent, the DEV should use separate counters for each key to maximize the number of secure frames that can be sent in that superframe. The secure frame counter is unique per DEV, so the DEV need only keep track of its own secure frame counter. For even better replay protection, a DEV may keep track of the last secure frame counter from any given DEV.**

223 (Gilb, TR) A 2-octet secure frame counter needs to be added to the secure frame formats in Figure 10, Figure 12, Figure 17 and Figure 19. The field should be called "Secure frame counter" and should be added directly after the time token in each figure. Add text to 7.3 that describes the secure frame counter field as follows: "The secure frame counter is used by the DEV for this frame to ensure uniqueness of the nonce." **Suggest accept.**

281 (Shvodian, TR) Secure Frame Counter (Data) or Sequence Counter (command) is missing. Not sure which one is used to protect the beacon. Add correct secure counter. **Suggest accept in principle. The secure frame counter should be included. Resolve as described in resolution to 223.**

434 (Gilb, TR) The integrity code needs a secure frame counter to operate correctly. Add a secure frame counter, 2 octets, to all secure frames at the beginning of the frame, right after the time token. Add the definition to 7.3, "The secure frame counter represents the number of times the slected key has been used during that superframe. The use of the secure frame counter in the encryption and integrity protocol is described in {xref}". **Suggest accept in principle.**

781 (Shvodian, TR) What does the Integrity code protect? Only the IEs or the SECID and secure sequence number, too? Clarify what the integrity code protects. The most important header fields are part of the nonce and thus already protected. **Suggest accept in principle. Figure 154 andTable 82 specify how to protect the beacon. Recommend referencing one of these in 7.3.1.2. Recommend that the SECID and secure**

frame counter not be protected by the integrity. Recommend also adding the following tables to clause 10.2.5 and, if desired, separating the entries of table 82 and interspersing these tables to help clarify:

3	2	6	1	octets: 1
Fragmentation control field	Secure frame counter	Time token	Destination DEVID	Source DEVID

Figure 2—CCM nonce format

Enc Data Length	Auth Data Length
0	10 + L_1 + ... + L_{n-1}

L_{n-1}	...	L_1	Octets: 10
Information element-(n-1)	...	Information element-1	Frame header

Figure 3—CCM input for secure beacons

Enc Data Length	Auth Data Length
L_2	14+L_1+L_2

L_2	L_1	2	2	Octets: 10
Pre-encrypted data	Authenticated data	Length (=4+L_1+L_2)	Command type	Frame header

Figure 4—CCM input for secure commands

Enc Data Length	Auth Data Length
L_1	10+L_1

L_1	Octets: 10
Pre-encrypted data	Frame header

Figure 5—CCM input for secure data frames

291 (Shvodian, TR) Need to show what the integrity code protects. Does it include SECID and sequence counter? **Suggest accept in principle. Resolve with comment 781**

935 (Shvodian, TR) I have been told that everything in the frame but the FCS is covered by the integrity check. There are some problems with this: HCS is not known during encryption so it cannot be part of integrity check. I recommended dropping HCS from the MAC andyway and keeping it in the PHY. A security wrapper is required to pad to a multiple of 16 octets. The calculation for this would have to include MAC overhead. Most of the important fields are protected by the nonce (SrcID, DestID, Fragmentation field, If possible, I think it is better if the integrity code does not cover the header. If it needs to, the covered fields need to made clear. HCS cannot be covered since it is generated at the PHY. **Suggest accept in principle. Resolve along with comment 781. Recommend that HCS not be covered by integrity code. Although it is redundant to include the 5 bytes that are in the nonce, recommend including the entire header (without the HCS) in the integrity code. The 5 bytes (SrcID, DestID, Frag. field) may be removed from being covered by the integrity code if desired. Padding is done internally in CCM and not added to the frame.**

493 (Gilb, TR) The IC needs to be recalculated if the frame is re-tried. Declare the retry bit to be a mutable field, i.e. that before calculating the IC, set this bit to a known value, say one. This is both for transmission and reception. **Suggest reject. Recommend requiring that a DEV re-protect the frame whenever retransmission is needed.**

769 (Shvodian, TR) Padding for security is needed. Security encrypts blocks of 128 bits (16 octets). Need to add padding for security, plus a field to indicate how many pad octets there are. **Suggest reject. CCM mode does not require the use of any padding and the encrypted text need not be a multiple of 16 octets.**

890 (Shvodian, TR) Padding needed to round up to 128 bit blocks. A mechanism is needed to pad the frame to a multiple of 16 octets and a way to indicate to the receiver how many octets of padding must be removed. May need a pad field in the secure frames. **Suggest reject. See recommendation for 769.**

1.1.8 Secure processing

873 (Shvodian, TR) It is not clear why the DEV would reject all commands while checking a message. Why wouldn't they be queued? Need to explain why commands are rejected. **Suggest accept in principle. The intent was to indicate that the DEV shall not process any commands while checking a message. If queuing is possible, this should be allowed.**

1.1.9 Frames

758 (Shvodian, TR) Does the length field in the Tx length include security overhead? What is covered by the length field needs to be clarified. **Suggest accept in principle. The Tx length should include the security overhead. See proposed frame format modification for comment 433. Recommend clarifying in the text in clause 7.2.**

567 (Gilb, TR) Need to describe how to receive an incoming secure frame. Add the following section to the end of 9.3 9.3.4 When a DEV receives a secure frame, it shall obtain the appropriate keying material from the MAC PIB depending on the SECID and source address found in the frame. To find the correct key, the DEV shall first check the MAC PIB for an ACL entry that corresponds to a peer-to-peer relationship with the sending DEV and that has a MACPIB_DataSECID or MACPIB_ManagementSECID that matches the received SECID. If no peer-to-peer ACL entry matches the received frame, the DEV shall check the MACPIB_PNCDataSECID and MACPIB_ManagementSECID to determine if it matches the received SECID. If either of these entries gives a match, the DEV shall use the security suite in the corresponding MACPIB_SecuritySuite and the key corresponding to the SECID. If an appropriate entry in the ACL cannot be found, the MLME shall return an MLME-SECURITY-ERROR.indication to the DME with the ReasonCode set to UNAVAILABLE-KEY and shall not perform any additional operations on the received frame. If the DEV is able to obtain the appropriate security suite and key from the ACL, the DEV shall compare the received time token to the value in the MACPIB_CurrentTimeToken. If the frame is a beacon frame, the DEV shall determine if the received time token is greater than the MACPIB_CurrentTimeToken. If the frame

is not a beacon frame, the DEV shall determine if the received time token is equal to the MACPIB_CurrentTimeToken. If either of these checks fail, the MLME shall return an MLME-SECURITY-ERROR.indication to the DME with the ReasonCode set to BAD-TIME-TOKEN and shall not perform any additional operations on the received frame. If the time token matches, the DEV shall apply the operations defined by the security suite to the frame. Before the security operations have been performed and the payload field has been modified, the DEV shall check the FCS. The DEV shall also check that the retry field in the frame control field of the MAC header is set to 0 and, if not, set it to 0. This operation is done in order to allow a device to retransmit a frame without recomputing the integrity code. The decryption operation shall be applied only to the integrity code, seeds that are being transmitted in a distribute key command or request key response command and the payload of data frames. The result of the decryption operation shall be replaced into the received frame in the place of the encrypted data. The integrity code shall be computed on the entire frame with the decrypted data replacing the encrypted data up to the integrity code itself including the MAC header. If any of the security operations fail, the MLME shall return an MLME-SECURITY-ERROR.indication to the DME with the ReasonCode set to FAILED-SECURITY-CHECK and shall not perform any additional operations on the received frame. If the security operations have been successfully performed and the frame has been modified appropriately, the device may then continue to process the frame. **Suggest accept in principle. Should discuss what informative text needs to be added and where to put it. Also should discuss the role of the PIB in other areas of the standard.**

566 (Gilb, TR) Need to have a desrciption of how to do the secure frame generation. Add the following subclause to 9.3 9.3.3 Secure frame generation When a DEV wishes to send a secure frame, it shall obtain the appropriate keying material from the MAC PIB depending on the key indicated by the DME. If the DME indicates that the PICONET-MGMT key shall be used, then the DEV shall use the key from the MACPIB_ManagementKeyInfo entry from the MAC PIB piconet security group parameters. If the DME indicates that the PICONET-DATA key shall be used, the DEV shall use the key from the MACPIB_DataKeyInfo entry from the MAC PIB piconet security group parameters. If the DME indicates that the PEER-MGMT key shall be used, the DEV shall use the key from the MACPIB_ManagementKeyInfo entry from the corresponding MAC PIB access control list group parameters table. If the DME indicates that the PEER-DATA key shall be used, then the DEV shall use the key from the MACPIB_DataKeyInfo entry from the corresponding MAC PIB access control list group parameters table. If the DEV is unable to find the corresponding key that is to be used, the MLME shall return an MLME-SECURITY-ERROR.indication to the DME with the ReasonCode set to UNAVAILABLE-KEY and shall not transmit the requested frame. If the MLME-xxx.request command has an associated MLME-xxx.confirm, then the MLME shall also set the reason code for the .confirm to be UNAVAILABLE-KEY. If the DEV is able to obtain the appropriate security suite and key from the MAC PIB, the DEV shall use the current time token in the frame. The SECID included in the frame shall be the value corresponding to the key being used. The integrity code shall be computed on the entire frame up to the integrity code itself including the MAC header. However, the DEV shall set the retry field in the frame control field of the MAC header to be 0 only for the purposes of the integrity calculation. This operation is done in order to allow a device to retransmit a frame without recomputing the integrity code. The result of the integrity code computation shall be encrypted and placed in the integrity code field in the secure frame. The encryption operation shall be applied only to the integrity code, seeds that are being transmitted in a distribute key command or request key response command and the payload of data frames. The result of the encryption operation shall be inserted into the frame in the place of the data that was encrypted. If any of the security operations fail, the MLME shall return an MLME-SECURITY-ERROR.indication to the DME with the ReasonCode set to FAILED-SECURITY-CHECK and shall not transmit the requested frame. If the security operations have been successfully performed and the payload field has been modified appropriately, the device shall then compute the FCS over the modified frame. **Suggest accept in principle. Should discuss along with 567.**

1.1.10 PNC Handover

336 (Heberling, TR) Figure 163 shows PNC handover using PNC information and PNC handover information (renamed to PNC handover CTRB). Non of these contains Authentication state. Consequently the new PNC has no way of knowing if a DEV is only associated, authenticated or in progress of authenticating. /KO

SEC group needs to clarify. Appropriate information elements needs to be added to PNC information, 7.5.4.2, or a new SEC handover command frame needs to be specified. Since the PNC information can be a response to a DEV inquiry, probably a new frame is the preferred alternative. **Suggest accept in principle. When the new PNC first become PNC, all devices become unauthenticated to the PNC until they authenticate with the new PNC. The new PNC shall allocate slots to authenticate each device in the piconet when it becomes the PNC. Recommend that a new command be added that may be sent by the departing PNC to transmit ACL information of all DEVs in the piconet to the new PNC and ACL information about the new PNC to all DEVs. This may ensure that the new PNC can more efficiently authenticate with each DEV in the piconet.**

1.1.11 ECC

Gilb (468, TR) The scheme in Annex B talks about a general elliptic curve, but 802.15.3 has chosen a specific one. Add an item here that defines the elliptic curve parameters, D, with a cross reference (xref 10.3.1.2). Also, use the nomenclature of Annex B.2 here (i.e. Hash, UID, VID, CAID, etc.) to better align the definitions with annex B. Probably reformat this as a table as well. **Suggest accept in principle. Recommend Struik provide updates to clarify the nomenclature used in Annex B.**

1.1.12 Diagrams & Figures

871 (Shvodian, T) This is unclear. Should be replaced by an MSC. **Suggest reject. There is already an MSC for PNC handover. Figure 160 is a state diagram for the security manager (including the PNC) for maintaining a secure piconet.**

459 (Gilb, TR) There is no pending key state in the diagram. Change "... to is the "pending key" state." to be "... to is the startup mode state or secure mode state." **Suggest accept in principle. The pending key state is shown in figure 160 and is the only state that may be transitioned to from startup mode that is not in the critical section. Should discuss how to clarify that the different state diagrams transition into each other.**

869 (Shvodian, TR) these states need to show an entry and exit. some only have one or the other. Show the state entry and exit. **Suggest reject. It is not possible to put all states in one diagram. In figure 159 (referenced in the comment) there is one incoming state (D0.5) and two outgoing states (SM0.0 and D0.0). The numbering of the states with a 0 at the beginning indicate that they come from the authentication diagrams. The states beginning with a 1 are key management related states.**

1.1.13 MLME messages

216 (Gilb, TR) Since the DME is able to choose the keys used for a command (or no keys), the .confirm commands need to add "UNAVAILABLE_KEY" to all of the result codes. Change all .confirm MLMEs that send frames as indicated. **Suggest accept in principle. If we accept 215 should we perhaps add more information to the MLME-SECURITY-ERROR.indication to indicate what command caused the error?**

215 (Gilb, TR) When devices are running in a secure mode, they need to be able to indicate to the DME when frames received or frames being sent cause security operation failures. These security operation failures could be caused by not having the specified key or by a failed integrity check or some other cryptographic failure. The following sub-clause should be added to Clause 6. 6.x.x Security management primitives These primitives define how the MLME communicates security related events to the DME. 6.x.x.x MLME-SECURITY-ERROR.indication This primitive allows the MLME of any DEV to indicate a failed security processing operation to the DME. The semantics of the primitive are as follows: MLME-SECURITY-ERROR.indication(SrcID, DestID, SECID, ReasonCode) The primitive parameters are defined in Table xx. Table xx - MLME-SECURITY-ERROR.indication parameters Name & Type & Valid Range & Description \\ SrcIDInteger & Any valid DEVID as defined in 7.2.3{xref} & The DEVID of the

entity from which the frame causing the error originated. \\ DestID & Integer & Any valid DEVID as defined in 7.2.3{xref} & The DEVID of the device for which the frame was intended. \\ SECID & Octet string & Any valid security session identifier. & Specifies the unique security session identifier for the key that was used on the incoming frame or that was requested to be used on the outgoing frame. \\ ReasonCode & Enumeration & UNAVAILABLE-KEY, FAILED-SECURITY-CHECK, BAD-TIME-TOKEN & The reason for the security error. \\ 6.x.x.x.x When generated This primitive is issued by the MLME when it receives an MLME.request message from a higher layer that requires security to be applied to a frame, but it is unable to find an appropriate key in the ACL or fails to be able to apply security to the frame. This primitive is also issued by the MLME when it receives a validly formatted frame from another device that induces a failed security check according to the security suite or for which the device is unable to find the designated key in the ACL. This primitive is also issued by the MLME when the time token received in a frame does not correspond to the current time token known by the DEV or if the last beacon was not valid. 6.x.x.x.x Effect on receipt On receipt of this primitive, the DME is notified of a security error and the reason for the security error. **Suggest accept in principle. Recommend review and accept text in 4.1.1 of 02/273r5.**

214 (Gilb, TR) Devices need to have the capability of choosing when to send frames with security and when not to. The decision for when to send a frame with security and what key to use should be determined by the DME.An indication needs to be added to each MLME.request and MLME.response in Clause 6, which cause the DEV to send a frame to another DEV, specifying whether that frame should be protected by security. Add the following parameter to the primtive descriptions for frames sent over the air. MLME-XXX.request (or .response)(KeySelection) with this entry in the corresponding tables. Name & Type & Valid range & Description \\ KeySelection & Enumeration & PICONET-MGMT, PICONET-DATA, PEER-MGMT, PEER-DATA, NONE & Specifies the key that shall be used to protect the outgoing frame or that security shall not be used on the frame. \\ **Suggest accept in principle. Recommend review and accept text in 4.1 of 02/273r5.**

213 (Gilb, TR) When the device is operating in security modes 1, 2 or 3, the MLME needs to be able to indicate to the DME what type of protection is used on a given received frame so that the DME can decide whether or not to accept the frame. This is important because some devices may want to choose to send unprotected frames to certain other devices and the DME needs to be able to determine whether its policy allows it to accept those frames. An indication needs to be added to each MLME.indication and each MLME.confirm in Clause 6, which indicates that a frame is received from another DEV, specifying whether the frame had security turned on and whether the frame came from a device in the ACL. The interfaces for the above described MLME messages should add the following entries to the semantics description: MLME-XXX.indication (or .confirm)(SecurityUse, ACLEntry) The following table entries should be added to the above described MLME messages. Name& Type & Valid Range & Description \\ SecurityUse & Boolean & TRUE or FALSE & This indicates to the DME if the received data frame had the security suite applied to it. \\ ACLEntry & Boolean & TRUE or FALSE & This indicates to the DME if the sender was found in the ACL. \\ **Suggest accept in principle. Recommend review and accept text in 4.1 of 02/273r5.**

1.1.14 Distribute key/Request key

868 (Shvodian, T) What is the purpose of the distribute Key response command? What does the PNC do if it fails? Try again? Disassociate the DEV? If the frame passed CRC the PNC would get an ACK and the key should be received correctly. Add text explaining the purpose of the distribute key response command. **Suggest accept in principle. The original purpose of the distribute key response was to give a cryptographic indication that the key was received. No action was indicated for what happens if the response was not received. Since secure ACKs in general are not considered to be necessary and since a device can always request the current key if it missed a distribute key command, recommend removing the distribute key response command.**

1112 (Shvodian, T) Confusion on reference sequence counter ... clause 9.9.4 has tow sequence counters. Which one is used. **Suggest accept in principle. The sequence counter replicates the sequence counter that is in the secure frame format (which is being deleted anyway), so it should be removed. The**

sequence counter (not the SFC) should be removed from the diagram in clause 9.9.4 as well, if the sequence counter is being deleted.

1111 (Roberts, T) Confusion on reference sequence counter ... clause 9.9.4 has tow sequence counters. Which one is used. Refer to security subcommittee for clarification. **Suggest accept in principle. Resolve along with 1112.**

1.1.15 Security modes

571 (Gilb, T) The description of security mode 0 is not descriptive enough and should refer to a DEV operating in the mode, not a piconet operating in the mode. Replacing the first paragraph in 9.4.1 with the following text: A device operating in security mode 0 shall not utilize the ACL entries and shall not perform any security related operations on MAC frames. While in this mode, if the MAC receives a frame with the SEC field set to 1, the MAC shall discard the frame and the MLME shall return an MLME-SECURITY-ERROR.indication to the higher layer with the ReasonCode set to UNAVAILABLE-KEY. If the MAC receives a frame with the SEC field set to 0, the MLME shall set the SecurityUse and ACLEntry fields to FALSE in the indication to the DME. **Suggest accept.**

572 (Gilb, T) The description of security mode 1 is not descriptive enough and should refer to a DEV operating in the mode, not a piconet operating in the mode. Replacing the first paragraph in 9.4.2 with the following text: Security mode 1 provides a mechanism for the MLME of a PNC to indicate to the DME if a received frame purportedly originated from a device in the ACL. The PNC may use this information as a criterion for allowing a device into the piconet. A device operating in security mode 1 shall not perform any security related operations on MAC frames. While in this mode, if the MAC receives a frame with the SEC field set to 1, the MAC shall discard the frame and the MLME shall return an MLME-SECURITY-ERROR.indication to the higher layer with the ReasonCode set to UNAVAILABLE-KEY. If the MAC receives a frame with the SEC field set to 0, the MLME shall set the SecurityUse field to FALSE and the ACLEntry field to TRUE or FALSE depending on if the sender is in the ACL in the indication to the higher layer. **Suggest accept.**

573 (Gilb, T) The description of security mode 2 is not descriptive enough and should refer to a DEV operating in the mode, not a piconet operating in the mode. Replacing the first paragraph in 9.4.3 with the following text: Security mode 2 provides a mechanism for a device to perform cryptographic security on frames transmitted in the piconet. DEVs operating in security mode 2 use public-key cryptography to verify the authenticity of other DEVs in the piconet and symmetric-key cryptography to protect frames using encryption and integrity. The cryptographic operations used for secure frames exchanged with the PNC and with other members of the piconet security group shall be performed as specified by the security suite indicated in the MACPIB_SecuritySuite in the MAC PIB piconet security group parameters. The cryptographic operations performed for secure frames exchanged with a peer DEV shall be performed as specified by the security suite indicated in the MACPIB_SecuritySuite entry associated with that peer security relationship in a MAC PIB access control list group parameters table. While in this mode, the MAC may accept frames with the SEC field in the frame control field set to 1 or 0 and shall set the SecurityUse in the MLME message to the DME to TRUE or FALSE respectively. **Suggest accept.**

574 (Gilb, T) The description of security mode 3 is not descriptive enough and should refer to a DEV operating in the mode, not a piconet operating in the mode. Replacing the first paragraph in 9.4.4 with the following text: Security mode 3 provides a mechanism for a device to perform cryptographic security on frames transmitted in the piconet. DEVs operating in security mode 3 use public-key cryptography and public-key certificates to verify the authenticity of other DEVs in the piconet and symmetric-key cryptography to protect frames using encryption and integrity. The cryptographic operations used for secure frames exchanged with the PNC and with other members of the piconet security group shall be performed as specified by the security suite indicated in the MACPIB_SecuritySuite in the MAC PIB piconet security group parameters. The cryptographic operations performed for secure frames exchanged with a peer DEV shall be performed as specified by the security suite indicated in the MACPIB_SecuritySuite entry associated with that peer

security relationship in a MAC PIB access control list group parameters table. While in this mode, the MAC may accept frames with the SEC field in the frame control field set to 1 or 0 and shall set the SecurityUse in the MLME message to the DME to TRUE or FALSE respectively. **Suggest accept.**

570 (Gilb, T) Need some more descriptive text for 9.4. The following descriptive text should be added to clause 9.4. The security mode indicates in what manner a DEV shall utilize the entries in the MAC PIB piconet security group parameter and MAC PIB access control list group parameters. The security mode in use is determined by the MACPIB_SecurityOptionImplemented entry in the MAC PIB. **Suggest accept.**

1.1.16 OIDs

882 (Shvodian, T) Are suite OIDs ever used, or do we just need subsuite OIDs? Eliminate suited OIDs if they serve no purpose. **Suggest accept in principle. The sub-suite OIDs are built off of the suite OID arcs. Since we are defining these OIDs for the first time in the standard, they should be included in the draft. The suite OIDs themselves are not used by the DEVs, though.**

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 6.

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

Figure 6—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 Figur e7.

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

Figure 7—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).
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

2.1 Monday, July 22, 2002

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.**

2.2 Tuesday, July 23, 2002

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.**

2.3 Friday, July 26, 2002

62 (Heberling, TR) Please clarify what impact the Security parameters have upon aMaxFrameSize-4? Does the amount of useful data get reduced to maintain the aMaxFrameSize-4? Please add clarification to the indicated sentence. **Accept in principle.** Add following sentence to end of clause 7.2.7, page 106, line 4: "When the SEC bit is set to 1, additional fields included in the frame body for payload protection will reduce the number of actual information octets by 14 {or 16 if SFC is included. See resolution of 888, 495, 891, 223, 281, 434}."

862 (Shvodian, T) Initial Owner needs a definition. Definine initial owner. **Accept in principle.** Add the following to 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), 885 (Shvodian, T), 886 (Shvodian, T), 887 (Shvodian, T), 892 (Shvodian, T): Defer to group resolution of 150.

463 (Gilb, TR), 847 (Shvodian, TR): **Accept in principle.** The term network byte order will be removed along with the need for the sequence counters.

870 (Shvodian, TR), 941 (Shvodian, T) **Accept in principle. Add 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 theAWAKE 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.

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}. Once all of the authenticated DEVs have been informed of the change, the PNC can change the SECID in the beacon. 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 once it sees the corresponding SECID in the beacon. 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 or entering the AWAKE state from the SLEEP 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-"the amount of time since the DEV last received a valid beacon with the known key"}.

224 (Gilb, TR), 846 (Shvodian, TR): **Accept in principle. Add following text to sub-clause 9.3 along with table to be added as part of resolution of comment 865.**

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 SECID for the piconet-wide group data key shall have the first octet set to the BcstID, {xref - 7.2.3}.The second octet shall designate a unique value for the key associated with the security relationship between the security manager and a DEV.

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. **Accept. Make sure definition and use of 'seed' is well defined.**

938 (Shvodian, T) For all of the secure frame formats, we may need to add the SMID (security manage ID) to the secure header so that a receiving DEV knows which key to use. Add the 8 bit SMID to security frames to enable a DEV to know which key was used for the encryltion. **Accept in principle. See resolution to 846.**

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. **Accept.**

927 (Shvodian, T) Secure ACK is not needed. Remove the Secure ACK message authentication generation. **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. **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. **Reject. No ACKs are shown on the diagrams since they do not add value to understanding the protocol, but do clutter up the diagram.**