IEEE P802.15
Wireless Personal Area Networks

Project	IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs)		
Title	**IEEE P802-15_TG3 MLME-ACL-INFO**		
Date Submitted	[September 12, 2002]		
Source	[Daniel V. Bailey, Ari Singer] [NTRU] [5 Burlington Woods Burlington, MA 01803 USA]	Voice: Fax: E-mail:	[+1 781 418-2522] [+1 781 418-2532] [dbailey@ntru.com]
Re:	802.15.3 TG3 Draft D11 for letter ballot 19		
Abstract	[This document is offered as a recommended change to 802.15.3 D11 for letter ballot 19.]		
Purpose	[This document is intended as support for ballot comments for letter ballot 19 on 802.15.3 D11.]		
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.		

Author’s note: Add the following sub-clause to 6.3 after 6.3.13. OIDs are currently sent on a per-device basis because a security manager may support multiple OIDs.

0.0.1 Retrieving ACL information

There primitives are used to request ACL information about other DEVs in the piconet. The parameters used for the MLME-ACL-INFO primitives are defined in Table 1.

Table 1—MLME-ACL-INFO primitive parameters

Name	Type	Valid range	Description
QueriedDEVID	Integer	Any valid DEVID as defined in 7.2.3.	Specifies the DEVID of the DEV for which information is being requested. A value of BcstID is defined as a request for information from all associated DEVs.
TrgtID	Integer	Any valid DEVID as defined in 7.2.3.	Specifies the DEVID of the DEV that the ACL information request is intended for.
OrigID	Integer	Any valid DEVID as defined in 7.2.3.	Specifies the DEVID of the DEV that initiated the MLME request.
ACLInfoSet	As defined in 0.0.2.2.	As defined in 0.0.2.2.	A set of ACL entry elements for the requested DEVs.
ACLInfoTimeout	Duration	0-65535	The time in milliseconds in which the operation initiated by the MLME request needs to be completed before responding with a ResultCode of TIMEOUT.
ResultCode	Enumeration	SUCCESS, TIMEOUT	Indicates the result of the MLME request.

0.0.1.1 MLME-ACL-INFO.request

This primitive initiates a request to the DEV for ACL information regarding either a single DEV or all of the DEVs in the piconet. The semantics of the primitive are as follows:

```
MLME-ACL-INFO.request          (
                               TrgtID,
                               QueriedDEVID,
                               ACLInfoTimeout
                               )
```

The primitive parameters are defined in Table 1.

0.0.1.1.1 When generated

The originating DME sends this primitive to its MLME when it wants to obtain ACL information about either an individual DEV or all of the DEVs in the piconet.

0.0.1.1.2 Effect of receipt

The MLME, upon receiving this primitive, sends the ACL information request command, 0.0.2.1, to the DEV specified by the TrgtID to request security information managed by that DEV.

0.0.1.2 MLME-ACL-INFO.indication

This primitive indicates the reception of a request by a DEV for ACL information it manages regarding either a specific DEV or all of the DEVs in the piconet. The semantics of the primitive are as follows:

```
MLME-ACL-INFO.indication        (
                                QueriedDEVID,
                                OrigID
                                )
```

The primitive parameters are defined in Table 1.

0.0.1.2.1 When generated

The DEV MLME sends this primitive to its associated DME upon receiving an ACL information request command, 0.0.2.1, from the requesting DEV specified by the OrigID.

0.0.1.2.2 Effect of receipt

The DME upon receiving this primitive sends an MLME-ACL-INFO.response to its MLME.

0.0.1.3 MLME-ACL-INFO.response

This primitive initiates a DME response to an MLME-ACL-INFO.indication. The semantics of the primitive are as follows:

```
MLME-ACL-INFO.response          (
                                OrigID,
                                ACLInfoSet
                                )
```

The primitive parameters are defined in Table 1.

0.0.1.3.1 When generated

The DME sends this primitive to its MLME as a result of receiving an MLME-ACL-INFO.indication.

0.0.1.3.2 Effect of receipt

The MLME upon receiving this primitive sends an ACL information command, 0.0.2.2, to the requesting DEV.

0.0.1.4 MLME-ACL-INFO.confirm

This primitive informs the originating DME its request for ACL information from the specified DEV is complete. The semantics of the primitive are as follows:

```
MLME-ACL-INFO.confirm           (
                                TrgtID,
                                ACLInfoSet,
                                ResultCode
                                )
```

The primitive parameters are defined in Table 1.

0.0.1.4.1 When generated

The MLME sends this primitive to its DME upon receiving either an ACL information command, 0.0.2.2, or a TIMEOUT.

0.0.1.4.2 Effect of receipt

The originating DME is informed whether its request for information about the either the single DEV or all of the DEVs in the piconet was successful or unsuccessful. If unsuccessful, the DME is able to resend the MLME-ACL-INFO.request. If successful, the DME will have acquired the information it requested.

Author’s note: Add the following sub-clauses to 7.5.

0.0.2 ACL information commands

The ACL information commands allow a DEV to request information about the public key of a specified DEV or DEVs. This command may be particularly useful following a PNC handover, after which each DEV authentiates with the new PNC. The new PNC may request information about the public keys trusted by the old PNC. Similarly, individual DEVs may request information about the public key of the new PNC.

0.0.2.1 ACL information request command

The DestID for the ACL information request command may be any valid DEVID of a peer DEV. The ACL information request command shall be formatted as illustrated in Figure 1.

octets: 1	**2**	**2**
Queried DEVID	Length (=1)	Command type

Figure 1—ACL information request command format

The queried DEVID is the DEVID of the DEV whose ACL information is being requested. If the value of this field is BcstID, then the DEV is requesting ACL information regarding the entire list of associated DEVs.

0.0.2.2 ACL information command

This command may be sent either as a response to the ACL information request by a DEV or it may be sent unsolicited. This command may be sent either in a directed command frame to a DEV or it may be sent in a broadcast command frame meant for all DEVs in the piconet. If the DestID is BcstID, then the ACK request field shall be no-ACK. The ACL information command shall be formatted as illustrated in Figure 2.

octets: L_m	**---**	**L_2**	**L_1**	**1**	**2**	**2**
DEV-m ACL record	…	DEV-2 ACL record	DEV-1 ACL record	Last	Length = $1+L_1+L_2+_{...}+L_m$	Command type

Figure 2—ACL information command

The last field is set to 0 if there are more DEV records that need to be transferred. The last field shall be set to 1 in the last command that is required for transferring the last of the ACL records.

The ACL record field shall be formatted as illustrated in Figure 3.

octets: L_2	**2**	**2**	**L_1**	**1**	**6**	**2**	**1**
Verification info	Verification info length = L_2	Verification info type	OID	OID length (= L_1)	DEV address	Length	DEVID

Figure 3—Format of an ACL record in an ACL information command

The DEVID is the ID assigned to the DEV by the PNC. This field shall not contain the broadcast or multicast IDs.

The DEV address is the MAC address of the DEV corresponding to the DEVID.

The OID length is the length of the OID corresponding to the preferred OID for that DEV. If this length is 0, no OID field shall be included.

The verification info type indicates the type of verification information that is included in the ACL entry. The valid verification info types are:

0 -> NULL
1 -> ECMQV Koblitz-283 key
2 -> RSA-OAEP raw 1024 key
3 -> NTRUEncrypt 251-1 key
4 -> ECMQV Koblitz-283 implicit certificate
5 -> X.509 certificate
6 -> X.509 CA certificate
7 -> ECMQV Koblitz-283 CA key
8 -> SHA-1 hash
9 -> SHA-256 hash
10 -> Certificate chain URL
11-255 -> Reserved

These types are defined in 0.0.4.

The verification info length indicates the length of the verification information that is included in the ACL entry. If this length is 0, no verification information field shall be included.

The verification info specifies the ACL verification info that may be used to verify the validity of the public key associated with that particular DEV.

Author’s note: Replace sub-clause 9.2.4 with the following sub-clause.

0.0.3 PNC handover

When a PNC chooses to handover the PNC role to another DEV in the piconet, the authentication relationships with the old PNC no longer apply to the new PNC. When the old PNC hands over the piconet information using a PNC information command, 7.5.4.2, the list of authenticated DEVs is passed to the new PNC.

PNC handover does not affect the group membership, so it does not require a rekey of the group keys. However, in a piconet with payload protection, the command functions of the PNC that relate to specific DEVs are not implemented until the new PNC has performed the authentication protocol with each DEV in the piconet. When the PNC role has been handed over, the new PNC should set up time slots for each of the authenticated DEVs to perform the authentication protocol with the new PNC.

The new PNC may request public-key verification information from the old PNC using an ACL information command, 10.0.2.2. Similarly, authenticated DEVs in the piconet may request public-key verification information from the old PNC using an ACL information command, 10.0.2.2. If the DME of each DEV chooses to accept this public-key verification information, the authentication process with each DEV may proceed without any interruption of service.

Author’s note: Add the following sub-clause at the end of 10.2.

0.0.4 Public-key verification information

DEVs may exchange public-key verification information that is intended to be used to verify public keys received in the authentication request command and challenge request command. The types of public-key verification information are specified in 0.0.2.2. The types have the following meanings:

Types 1 through 5 correspond to the actual public key objects themselves.

Type 6, X.509 CA certificate, corresponds to an X.509 certificate as defined in RFC 3280 that belongs to the CA that signed the corresponding DEV’s certificate.

Type 7, ECMQV Koblitz-283 CA key, corresponds to the ECC public key of the CA that signed an implicit certificate for the ecmqv-implicit-1 sub-suite.

Type 8 , SHA-1 hash, is used to transmit the hash of the public-key and ID used in the rsa-oaep-raw-1 and ntruencrypt-raw-1 sub-suites.

Type 9 , SHA-256 hash, is used to transmit the hash of the public-key and ID used in the ecmqv-manual-1 sub-suite.

Type 10, certificate chain url, is used to transmit a uniform resource locator at which the certificate chain from which a certificate is built.