Freescale Semiconductor
Application Note

Document Number: AN4973
Rev. 0, 7/2014

Freescale 802.15.4 MAC Security Setup

MAC 2006 and MAC 2011 Group Key Security Examples

By: Alexandru Balmus, Embedded Connectivity Software Engineer, Bucharest, Romania

The 802.15.4 MAC is a standard for Low Rate Wireless Personal Area Networks used for applications in various fields: building automation, utility metering, medical devices, etc. The 802.15.4 MAC is used as a lower layer by multiple network layer standards each targeted at specific applications (ZigBee, RF4CE, ZigBee IP, 6LoWPAN).

The purpose of this application note is to thoroughly explain how to configure MAC Security for a very common use case.

Contents

1 Introduction

One of the most common scenarios of security use in 802.15.4 wireless networks is the following:

An application level master device must securely communicate with multiple peripheral devices. A

peripheral device communicates only with the central and not with other peripheral devices. Only one key is used by all the devices.

This use case maps well over the 802.15.4 MAC network topology, which is a star type topology. The network is composed of a Coordinator and multiple End Devices. The following diagram summarizes the use case.

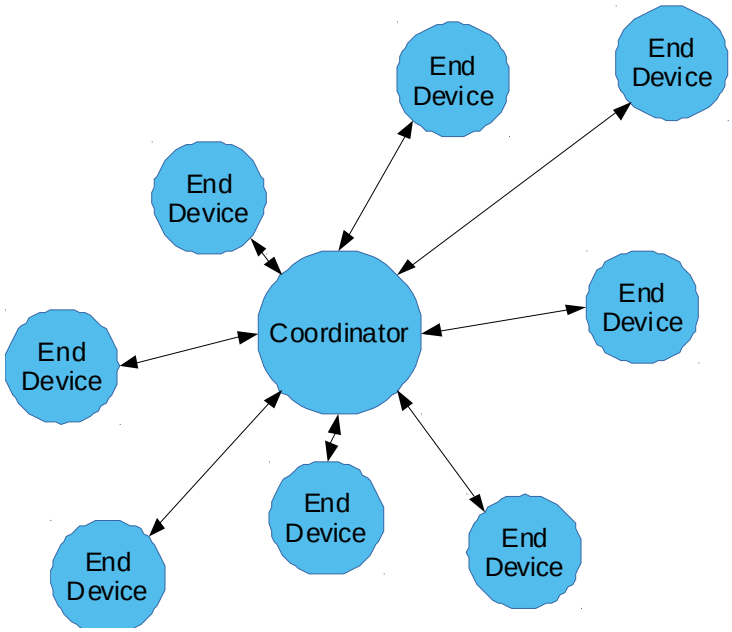

Figure 1. Secured Network — Central Device communicates with multiple peripherals

2 802.15.4 MAC Security

The 802.15.4 MAC offers the option to encrypt and/or authenticate frames. The MAC security mechanisms also provide replay protection. Both MAC Data Frames (MCPS) and MAC Command Frames (MLME) can be secured.

MAC security operations (encryption and authentication) are based on the AES (Advanced Encryption Standard) block cypher — specifically AES-128, which works with a block and key size of 128 bits. The cypher is not used directly but through a variation of the Counter with CBC-MAC mode (CCM) called CCM*. CCM always operates in the same manner: it encrypts and adds a signature (MIC — MAC Integrity Check) to the input data. In contrast to CCM, the CCM* mode offers the option to encrypt only or only add the signature to the input data without encryption. The mode of operation of CCM* is selected by using the SecurityLevel parameter, which is given as input to all MCPS-DATA.requests to the MAC. The summary of the MAC security levels and their effects on the frames are shown in the following table.

Table 1. MAC security levels and their effects on the frames

Security Level	Name	Encryption	MIC Length
0x00	N/A	No	0
0x01	MIC-32	No	4
0x02	MIC-64	No	8
0x03	MIC-128	No	16
0x04	ENC	Yes	0
0x05	ENC-MIC-32	Yes	4
0x06	ENC-MIC-64	Yes	8
0x07	ENC-MIC-128	Yes	16

Securing frames increases power consumption and increases the time needed to send frames, thus decreasing throughput. Throughput is also decreased while using MAC security because the length of the frame useful payload is decreased due to the inclusion of the MIC (as seen above) and the Auxiliary Security Header (ASH). The length of the ASH depends on the KeyIdMode (Key Identifier Mode) parameter. The following table shows the content and length of the ASH based on KeyIdMode.

Table 2. Content and length of the ASH based on KeyIdMode

Key Id Mode	ASH Fixed Part	ASH Key Identifier Field	ASH Total Length
0x00	Security Control [1] + Frame Counter [4] = 5	—	5 + 0 = 5
0x01	Security Control [1] + Frame Counter [4] = 5	Key Index [1] = 1	5 + 1 = 6
0x02	Security Control [1] + Frame Counter [4] = 5	Key Index [1] + Key Source [4] = 5	5 + 5 = 10
0x03	Security Control [1] + Frame Counter [4] = 5	Key Index [1] + Key Source [8] = 9	5 + 9 = 14

To use security with the 802.15.4 MAC, some PIB (PAN Information Base) attributes must be properly configured. Depending on the security mode used, these include general MAC PIB attributes and MAC security PIB attributes. The most important attributes are the security-related PIB attributes, which contain the Security Tables (Key Table (Key Id Lookup List, Key Device List, Key usage List), and Security Level Table, Device Table).

The main purposes of the MAC Security Tables are:

- Determining the key for securing an outgoing frame
- Determining the key for unsecuring an incoming frame
- Determining whether a type of frame can be secured/unsecured with a certain key
- Determining whether a frame meets the minimum security requirements
- Determining whether a peer device is allowed to send secured frames

The Security Tables offer great flexibility in using the MAC but configuring them is a rather complex task. Their configuration for the described use scenario is detailed in the following sections.

2.1 Addressing Modes and Security

All 802.15.4 devices have an unique 64 bit (8 byte) address called Extended Address. Devices can communicate using the Extended Address or a 16-bit (2-byte) address called Short Address.

The Extended Address is usually hardcoded — constant aExtendedAddress for the 2006 version of the MAC and macExtendedAddress read-only PIB attribute for the 2011 version of the MAC. The Freescale implementation offers the option to override this address using a PIB Attribute for both MAC 2006 and MAC 2011.

The Short Addresses on the other hand are allocated by the Coordinator to End Devices during the MAC Association Procedure. The values of the short addresses are allocated and managed by the Application layer of the Coordinator.

The Extended Addresses and Short Addresses of devices are used in the MAC security tables and it is the Application's job to configure and reconfigure (if needed) the correct addresses and addressing modes in the security tables.

To properly use addressing modes and MAC security, an application must do the following:

- Configure correct addresses and addressing modes in the security tables — extended or short
- Use addresses and addressing modes in the MAC Data requests that match the addresses and addressing modes set up in the security tables
- Change the address and/or addressing mode in the security tables if a device's short address is allocated/deallocated/reallocated or changed

2.2 MAC MCPS-DATA.request and Security Parameters

The security parameters in MAC Data Requests must match the parameters written in the MAC security tables for the operations of securing and unsecuring frames to be successful.

The following table summarizes the parameters of the MCPS-DATA.request that are relevant for security operations.

Table 3. Parameters of the MCPS-DATA.request relevant for security operations

MCPS-DATA.request Parameter	Description
SrcAddrMode	Source addressing mode — extended addressing or short addressing
SrcPANId	Source PAN Identifier. Freescale proprietary parameter — can override the PAN Id set in the MAC PIB attributes
SrcAddr	Source address — extended or short, as specified by SrcAddrMode. Freescale proprietary parameter — can override the addresses set in the MAC PIB attributes
DstAddrMode	Destination addressing mode - extended addressing or short addressing
DstPANId	Destination PAN Identifier.
DstAddr	Destination address — extended or short, as specified by DstAddrMode.

Table 3. Parameters of the MCPS-DATA.request relevant for security operations (continued)

MCPS-DATA.request Parameter	Description
SecurityLevel	Security level. Specifies how the frame is secured using AES-CCM*
KeyIdMode	Key Identifier Mode. One byte attribute determining the format of the Auxiliary Security Header and how the proper key for securing/unsecuring frames is searched for in the security tables
KeySource	Key Source. 4- or 8-byte attribute determining the originator of a group key. Will contain a PANId concatenated with a ShortAddress if it is 4 bytes in length and an ExtendedAddress if it is 8 bytes in length. If the KeyIdMode parameter equals 0 or 1 this parameter is not used.
KeyIndex	Key Index. One byte attribute used to differentiate between multiple keys sharing KeySource information. Can have any value.

3 Freescale MAC Security Tables

The 802.15.4 MAC standard allows additional implementation specific constraints on read-write operations of the security-related PIB attributes.

To save memory the sizes of the Freescale 802.15.4 MAC security tables are set at compile time and cannot be changed dynamically at runtime. Thus, the PIB attributes for the actual security tables and the PIB attributes for the security tables sizes are read only. Example: you cannot write the gMPibKeyTable_c = 0x71 and the gMPibKeyTableEntries_c = 0x72 MAC PIB attributes for neither the bare metal nor the RTOS based Freescale 802.15.4 MAC stacks.

To write the contents of the security tables the Freescale 802.15.4 MAC implementation offers two types of implementation specific attributes: table entry index attributes for tables and subtables (e.g. gMPibKeyTableCrtEntry_c = 0x96, gMPibKeyUsageListCrtEntry_c = 0x9A, gMPibDeviceTableCrtEntry_c = 0x97, etc.) and attributes for writing elements of tables and subtables (e.g. gMPibKey_c = 0x85 - the Key element of a KeyDescriptor from the KeyTable, gMPibKeyIdLookupData_c = 0x94 - the KeyIdLookupData element of the KeyIdLookupDescriptor from the KeyIdLookupList).

To write an element of a security table or subtable the index attributes for that table and subtable must be set up properly. For example to write the FrameType element of a KeyUsageDescriptor from the KeyUsageList included into a KeyDescriptor from the KeyTable three MLME Set PIB Attribute operations must be done:

- Write the gMPibKeyTableCrtEntry_c = 0x96 attribute with the index of the KeyDescriptor in the KeyTable
- Write the gMPibKeyUsageListCrtEntry_c = 0x9A attribute with the index of the KeyUsageDescriptor in the KeyUsageList
- Write the gMPibKeyUsageFrameType_c = 0x86 attribute with the value desired for the FrameType element

The actual written element after the above Set PIB operations will be:

KeyTable[gMPibKeyTableCrtEntry_c].KeyUsageList[gMPibKeyUsageListCrtEntry_c].FrameType

This implementation choice was made to minimize the MLME Set PIB operations memory usage when the security tables are very large. Multiple keys require multiple KeyDescriptors in the KeyTable, multiple devices which use a key require multiple KeyDeviceDescriptors in the KeyDeviceList of a particular KeyDescriptor.

NOTE

The names of the table index and table elements PIB attributes preprocessor definitions may be slightly different for various versions of the MAC but the PIB numbers will be the same.

3.1 Security Tables Configuration

The following table contains the locations and the names of the files where the C Preprocessor definitions for the MAC security tables' sizes can be found and configured.

Table 4. Locations and names of files where C Preprocessor definitions for MAC security tables' sizes

MAC Ver.	Bare metal MAC	MQXLite RTOS based MAC
File	AppToMacPhyConfig.h	MacGlobals.h
Location (may vary)	<ProjectFolder>\Application\Configure\	<StackRootFolder>\ieee_802_15_4\Source\App\
Note	Default location in BeeKit generated projects	

The actual sizes of the MAC security tables must be assigned to the macro definitions in the tables below, for the bare metal and RTOS based MAC versions. The default values for all tables' sizes is 2.

Table 5. Macro definitions

File	Bare metal MAC
	AppToMacPhyConfig.h
Code	#define gNumKeyTableEntries_c 2 #define gNumKeyIdLookupEntries_c 2 #define gNumKeyDeviceListEntries_c 2 #define gNumKeyUsageListEntries_c 2 #define gNumDeviceTableEntries_c 2 #define gNumSecurityLevelTableEntries_c 2

Table 5. Macro definitions (continued)

	MQXLite RTOS based MAC
File	MacGlobals.h

Code	`#define gNumKeyTableEntries_c 2`
	
	`#define gNumKeyIdLookupListEntries_c 2`
	`#ifndef gMAC2011_d`
	`  #define gNumKeyDeviceListEntries_c 2`
	`#else /* gMAC2011_d */`
	`  #define gNumDeviceDescriptorHandleListEntries_c 2`
	`#endif /* gMAC2011_d */`
	`#define gNumKeyUsageListEntries_c 2`
	
	`#define gNumDeviceTableEntries_c 2`
	`#define gNumSecurityLevelTableEntries_c 2`

NOTE

The table sizes for the KeyIdLookupList, KeyDeviceList and KeyUsageList, which are subtables of the KeyTable, apply for each element (KeyDescriptor) of the KeyTable

4 Use scenario

As described in the Introduction of this document, the use case involves a central device communicating with multiple peripheral devices using a single key.

Use case summary:

- n + 1 devices
  - 1 central, n peripherals
  - Peripheral device: 1 entry in the KeyDeviceList of the KeyDescriptor and 1 entry in the DeviceTable — for the central device
  - Central device: n entries in the KeyDeviceList of the KeyDescriptor and n entries in the DeviceTable — for each peripheral device
- 1 key
  - 1 entry in the KeyTable of each device - a single KeyDescriptor
- KeyIdMode = 1
  - 1 entry in the KeyIdLookupList - a single KeyIdLookupDescriptor
  - This KeyIdMode must be used in every MCPS Data Request
  - The KeyIdIndex must be set up correctly in the LookupData element of the KeyIdLookupDescriptor
  - For this KeyIdMode the AuxiliarySecurityHeader sent over the air will have the following format: ASH = SecurityControl[1] + FrameCounter[4] + KeyIdentifier[1](KeyIndex[1]) = 6 octets
- SecurityLevel = 6

  - The data is encrypted and a MIC64 is added to the frame
  - This SecurityLevel must be used in every MCPS Data Request
- Only data frames are secured
  - 1 entry in the SecurityLevelTable for data frames — a single SecurityLevelDescriptor
  - 1 entry in the KeyUsageList of the KeyDescriptor — a single KeyUsageDescriptor

NOTE

For this security use scenario where KeyIDMode = 1 the KeySource parameter from the MCPS Data Request is not used and can have any value.

For this security use scenario the SecurityLevel, KeyIdMode and KeyIndex parameters must be properly set in each MCPS Data Request

4.1 MAC 2006 PIB Configuration

The following table shows how to set up the key table sizes for the described scenario for the central (C) and peripheral (P) devices for the 2006 version of the 802.15.4 MAC. These settings are valid for both the bare metal and RTOS based MAC versions.

Table 6. MAC 2006 — How to set up the key table sizes

Security Table Size Macro Definition	C	P	Notes
gNumKeyTableEntries_c	1	1	Equal to the number of keys used by a device. In this scenario only one key is used by all devices.
gNumKeyIdLookupListEntries_c	1	1	One entry for each KeyIdMode used for a specific key.
gNumKeyDeviceListEntries_c	n	1	One entry for every device from which secured frames must be received.
gNumKeyUsageListEntries_c	1	1	One entry for every secured frame type. In this scenario only data frames are secured
gNumDeviceTableEntries_c	n	1	One entry for every device from which secured frames must be received.
gNumSecurityLevelTableEntries_c	1	1	One entry for every secured frame type. In this scenario only data frames are secured

The table below shows how to set up all the necessary the MAC PIB attributes and in what order for the given scenario and some additional information.

Table 7. MAC 2006 — How to set up all the necessary the MAC PIB attributes (Sheet 1 of 4)

General MAC PIB Attributes	PIB attributes that are not necessarily security related	
gMPibSecurityEnabled_c	0x01	Global MAC Security Enable switch
gMacPibExtendedAddress_c	0xFFEEDDCCBBAA9988	Device long address
gMPibShortAddress_c	0xCAFE	Device short address
gMPibPanId_c	0x1AAA	PAN Id used in the network

Table 7. MAC 2006 — How to set up all the necessary the MAC PIB attributes (Sheet 2 of 4)

Security MAC PIB Attributes	PIB attributes which are not necessarily security related	
gMPibDefaultKeySource_c	0x0011223344556677	Common default key source for all devices
gMPibFrameCounter_c	0x00000000	Or any value needed/recovered from NVM. Number depends on the number of secured frames sent by a device.
Security MAC PIB Attributes – Key Table	**One entry (KeyDescriptor) for each key used by a device**	
gMPibiKeyTableCrtEntry_c	0	**Index 0** - Index of the KeyDescriptor in the KeyTable
gMPibKey_c	0x8899AABBCCDDEEFF	The key used by all devices
Security MAC PIB Attributes – Key Table – Key ID Lookup List	**One entry (KeyIdLookupDescriptor) for every KeyIdMode used for a specific key**	
gMPibiKeyIdLookuplistCrtEntry_c	0	**Index 0 -** Index of the KeyIdLookupDescriptor in the KeyIdLookupList
gMPibKeyIdLookupData_c	macPibDefaultKeySource [8] || keyIndex [1]	For KeyIdMode = 1 – used in the data request The LookupData element has a different format for each KeyIdMode. Different PIB attributes may need to be set up for each KeyIdMode.
gMPibKeyIdLookupDataSize_c	1	For KeyIdMode = 1 - used in the data request - size is 9 octets The LookupDataSize element has a different value for each KeyIdMode
Security MAC PIB Attributes – Security Level Table	**One entry (SecurityLevelDescriptor) for every secured frame type**	
gMPibiSecurityLevelTableCrtEntry_c	0	**Index 0** - Index of the SecurityLevelDescriptor in the SecurityLevelTable
gMPibSecLevFrameType_c	0x01	data frame
gMPibSecLevCommnadFrameIdentifier_c	0x00	Irrelevant for this frame type
gMPibSecLevSecurityMinimum_c	0x06	Or whatever security level is used in the MCPS Data Request
gMPibSecLevDeviceOverrideSecurityMinimum_c	TRUE	Or FALSE – Depending on what application needs
Repeat the Above 5 Set PIBS to Add More Frame Types – With a Different Index Each		

Table 7. MAC 2006 — How to set up all the necessary the MAC PIB attributes (Sheet 3 of 4)

Security MAC PIB Attributes – Key Table – Key Device List	**One entry (KeyDeviceDescriptor) for each device from which the current device must receive secured frames. If multiple entries are present in the KeyDeviceList, the UniqueDevice element must be set to FALSE for all entries**	
gMPibiKeyDeviceListCrtEntry_c	i	**Index i** – Index of the KeyDeviceDescriptor in the KeyDeviceList
gMPibKeyDeviceDescriptorHandle_c	i	Index in the DeviceTable for the current KeyDeviceList entry. The DeviceDescriptorHandle element must point to the appropriate index in the DeviceTable.
gMPibUniqueDevice_c	FALSE	Multiple devices defined in the DeviceList use this key
gMPibBlackListed_c	FALSE	Automatically set to TRUE by the MAC when the FrameCounter reaches 0xFFFFFFFF
Repeat the Above 4 Set PIBS to Add More Key Device List Entries for Each Device that Uses the Current Key. Each Entry Should Have a Different Device Descriptor Handle		
Security MAC PIB Attributes – Key Table - Key Usage List	**One entry for every secured frame type**	
gMPibiKeyUsageListCrtEntry_c	0	**Index 0** - Index of the KeyUsageDescriptor in the KeyUsageList
gMPibKeyUsageFrameType_c	0x01	Data frame type
gMPibKeyUsageCommnadFrameIdentifier_c	0x00	Irrelevant for this frame type
Repeat the Above 3 Set PIBS to Add More Frame Types – with a Different Index Each		
Security MAC PIB Attributes – Device Table	**One entry for each device from which the current device must to receive secured frames**	
gMPibiDeviceTableCrtEntry_c	i	Index i
gMPibDeviceDescriptorPanId_c	0x1AAA	PAN Id used in the network
gMPibDeviceDescriptorShortAddress_c	0x0000	Short address of the device from which the frame is received (if short addressing is used in the data requests)

Table 7. MAC 2006 — How to set up all the necessary the MAC PIB attributes (Sheet 4 of 4)

gMPibDeviceDescriptorExtAddress_c	0x1122334455667788	Extended address of the device from which the frame is received (if extended addressing is used in the data requests)
gMPibDeviceDescriptorFrameCounter_c	0x00000000	Expected FrameCounter of this device. Automatically updated by the Mac as it receives secured frames from a device.
gMPibDeviceDescriptorExempt	TRUE	May be FALSE - depending on application needs
Repeat the Above 6 Set PIBS to Add More Devices – with a Different Index and Different Extended Address, Short Address and Frame Counter		

4.2 MAC 2011 PIB Configuration

The following table shows how to set up the key table sizes for the described scenario for the central (C) and peripheral (P) devices for the 2011 version of the 802.15.4 MAC. There is only a RTOS based MAC 2011 version.

Table 8. MAC 2011 — How to set up the key table sizes

Security Table Size Macro Definition	C	P	Notes
gNumKeyTableEntries_c	1	1	Equal to the number of keys used by a device. In this scenario only one key is used by all devices.
gNumKeyIdLookupListEntries_c	1	1	One entry for each KeyIdMode used for a specific key.
gNumDeviceDescriptorHandleListEntries_c	n	1	One entry for every device from which secured frames must be received.
gNumKeyUsageListEntries_c	1	1	One entry for every secured frame type. In this scenario only data frames are secured
gNumDeviceTableEntries_c	n	1	One entry for every device from which secured frames must be received.
gNumSecurityLevelTableEntries_c	1	1	One entry for every secured frame type. In this scenario only data frames are secured

The MAC PIB attributes which contain the sizes of the security tables and subtables (gMPibKeyTableEntries_c = 0x72, gMPibDeviceTableEntries_c = 0x74, MPibSecurityLevelTableEntries_c = 0x76, gMPibKeyIdLookupListEntries_c = 0x80, gMPibKeyUsageListEntries_c = 0x84, gMPibDeviceDescriptorHandleListEntries_c = 0x92) were removed from the 2011 version of the 802.15.4 MAC standard. For backwards compatibility and for the memory saving related reasons mentioned in the previous paragraphs these attributes were kept in the Frescale implementation. As mentioned before additional constraints are allowed by the standard for security-related PIB attributes.

For each security table and subtable, the Freescale MAC 2011implementation offers indexing PIB attributes which must be properly set before writing an element of a table or subtable. How to use these index PIB attributes was described in the previous sections.

The table below shows how to set up all the necessary the MAC PIB attributes and in what order for the given scenario and some additional information.

Table 9. MAC 2011 — How to set up all the necessary the MAC PIB attributes (Sheet 1 of 3)

General MAC PIB Attributes	PIB attributes that are not necessarily security related	
gMPibSecurityEnabled_c	0x01	Global MAC Security Enable switch
gMacPibExtendedAddress_c	0xFFEEDDCCBBAA9988	Device long address
gMPibShortAddress_c	0xCAFE	Device short address
gMPibPanId_c	0x1AAA	PAN Id used in the network
Security MAC PIB Attributes	**PIB attributes that are not necessarily security related**	
gMPibDefaultKeySource_c	0x0011223344556677	Common default key source for all devices
gMPibFrameCounter_c	0x00000000	Or any value needed/recovered from Non Volatile Memory. Number depends on the number of secured frames sent by a device.
Security MAC PIB Attributes – Key Table	**One entry (KeyDescriptor) for each key used by a device**	
gMPibiKeyTableCrtEntry_c	0	**Index 0** - Index of the KeyDescriptor in the KeyTable
gMPibKey_c	0x8899AABBCCDDEEFF	The key used by all devices
Security MAC PIB Attributes – Key Table – Key ID Lookup List	**One entry (KeyIdLookupDescriptor) for every KeyIdMode used for a specific key**	
gMPibiKeyIdLookuplistCrtEntry_c	0	**Index 0 -** Index of the KeyIdLookupDescriptor in the KeyIdLookupList
gMPibKeyIdLookupKeyIdMode_c	1	The KeyIdMode used in the MCPS Data Request
gMPibKeyIdLookupKeyIndex_c	ki	The KeyIndex used in the MCPS Data Request
gMPibKeyIdLookupKeySource_c	0x0011223344556677	For KeyIdMode = 1 this value must be equal to the gMPibDefaultKeySource_c PIB attribute
The Rest of the Elements of the Key ID Lookup Descriptor Are Not Used for Key ID Mode 1		

Table 9. MAC 2011 — How to set up all the necessary the MAC PIB attributes (Sheet 2 of 3)

Security MAC PIB Attributes – Security Level Table	One entry (SecurityLevelDescriptor) for every secured frame type	
gMPibiSecurityLevelTableCrtEntry_c	0	**Index 0** - Index of the SecurityLevelDescriptor in the SecurityLevelTable
gMPibSecLevFrameType_c	0x01	data frame
gMPibSecLevCommnadFrameIdentifier_c	0x00	Irrelevant for this frame type
gMPibSecLevSecurityMinimum_c	0x06	Or whatever security level is used in the MCPS Data Request
gMPibSecLevDeviceOverrideSecurityMinimum_c	TRUE	Or FALSE – Depending on what application needs
gMPibSecLevAllowedSecurityLevels_c	0xFF, … , 0xFF	List of maximum 8 security levels allowed for this frame type. Set all to 0xFF for this use case (all values 0xFF = empty list).
Repeat the Above 6 Set PIBS to Add More Frame Types – with a Different Index Each		
Security MAC PIB Attributes – Key Table – Device Descriptor Handle List	**One entry (DeviceDescriptorHandle) for each device from which the current device must receive secured frames.**	
gMPibiDeviceDescriptorHandleListCrtEntry_c	i	**Index i** – Index of the DeviceDescriptorHandle in the DeviceDesciptorHandleList
gMPibDeviceDescriptorHandle_c	i	Index in the DeviceTable for the current DeviceDesciptorHandleList entry.
Repeat the Above 2 Set PIBS to Add More Device Descriptor Handle List Entries for Each Device that Uses the Current Key.		
SECURITY MAC PIB ATTRIBUTES – KEY TABLE - KEY USAGE LIST	**One entry for every secured frame type**	
gMPibiKeyUsageListCrtEntry_c	0	**Index 0** - Index of the KeyUsageDescriptor in the KeyUsageList
gMPibKeyUsageFrameType_c	0x01	Data frame type
gMPibKeyUsageCommnadFrameIdentifier_c	0x00	Irrelevant for this frame type
Repeat the Above 3 Set PIBS to Add More Frame Types – with a Different Index Each		
Security MAC PIB Attributes – Device Table	**One entry for each device from which the current device must to receive secured frames**	
gMPibiDeviceTableCrtEntry_c	i	Index i

Table 9. MAC 2011 — How to set up all the necessary the MAC PIB attributes (Sheet 3 of 3)

gMPibDeviceDescriptorPanId_c	0x1AAA	PAN Id used in the network
gMPibDeviceDescriptorShortAddress_c	0x0000	Short address of the device from which the frame is received (if short addressing is used in the data requests)
gMPibDeviceDescriptorExtAddress_c	0x1122334455667788	Extended address of the device from which the frame is received (if extended addressing is used in the data requests)
gMPibDeviceDescriptorFrameCounter_c	0x00000000	Expected FrameCounter of this device. Automatically updated by the MAC as it receives secured frames from a device.
gMPibDeviceDescriptorExempt	TRUE	May be FALSE – depending on application needs
Repeat the Above 6 Set PIBS to Add More Devices – with a Different Index and Different Extended Address, Short Address and Frame Counter		

5 Application Summary

This section summarizes how an application using the Association procedure should work with MAC security. An application draft is shown for both a Coordinator and EndDevice. For full examples with code, please see the MyStarNetwork and MyWirelessApp demo applications that are provided with the MAC Software Stack.

5.1 Coordinator Application

1. Platform initialization
2. Stack initialization
3. Application initialization
4. Start the PAN as Coordinator
5. Security initialization
   a.) Set up General MAC PIB attributes related to security
   b.) Set up the entry in the Key Table
      i.) Set up the key
      ii.) Set up the Key Id lookup List
      iii.) Set up the Key Usage List
   c.) Set up the Security Level Table
6. If an association request is received from an EndDevices
   a.) If the association procedure is successful (with short or extended address)

i.) Set up an entry with correct addressing settings in the Device Table for the associated End Device

ii.) Set up an entry for the associated EndDevice in the Key Device List of the KeyDescriptor in the Key Table - with a Device Descriptor Handle pointing to the corresponding index in the entry created in the Device Table

7. Send and receive secured frames to and from the associated EndDevices

5.2 EndDevice Application

1. Platform initialization
2. Stack initialization
3. Application initialization
4. Scan for a PAN Coordinator
5. If a proper Coordinator is found start the association procedure
6. If the association procedure is successful perform security setup
   a.) Set up General MAC PIB attributes related to security
   b.) Set up the entry in the Key Table
      i.) Set up the key
      ii.) Set up the Key Id lookup List
      iii.) Set up the Key Usage List
   c.) Set up the Security Level Table
   d.) Set up an entry with correct addressing settings in the Device Table for the Coordinator
   e.) Set up an entry for the Coordinator in the Key Device List of the KeyDescriptor in the Key Table - with a Device Descriptor Handle pointing to the corresponding index in the entry created in the Device Table
7. Send and receive secured frames to and from the Coordinator

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Document Number: AN4973
Rev. 0
7/2014