6TiSCH
INTERNET-DRAFT
Intended Status: Informational
Expires: June 13, 2015

G. Piro
(Politecnico di Bari)
G. Boggia
(Politecnico di Bari)
L. A. Grieco
(Politecnico di Bari)

December 10, 2014

Layer-2 security aspects for the IEEE 802.15.4e MAC
draft-piro-6tisch-security-issues-03

Abstract

The aim of this Internet Draft is to define standard compliant procedures for configuring layer-2 security services in IEEE 802.15.4e-based Low-power and Lossy Networks. In particular, it provides a review of security aspects presented in both IEEE 802.15.4-2011 and IEEE 802.15.4e-2012 specifications, the classification of secure network configurations and layer-2 keys, the description of a set of consecutive steps required to establish a layer-2 secure link, and a lightweight Key Management Protocol designed for negotiating a layer-2 one-hop link key. As the final goal, the document would describe how security MAC attributes can by initialized and updated in order to offer layer-2 security services in real networks.

http://www.ietf.org/shadow.html

Table of Contents

1 Acronyms

In addition to the acronyms defined in [I-D.ietf-6tisch-terminology], the following acronyms are used in this document:

ECDH Elliptic Curve Diffie Hellman

KMP Key Management Protocol

2 Introduction

The IEEE 802.15.4 standard [IEEE802154] is widely recognized as one of the most successful enabling technologies for short-range low-rate wireless communications. It covers all the details related to the Medium Access Control (MAC) and physical layers of the protocol stack and supports the possibility to protect MAC packets by means of symmetric-key cryptography techniques with several security options. However, the IEEE 802.15.4 standard does not explain how handling the initialization of a secure IEEE 802.15.4 domain, the generation and the exchange of keys, and the management of joining operations in a secure 802.15.4 network already configured in the past, thus delegating the upper layers to orchestrate, enable, configure, and negotiate security services. The IEEE 802.15.4e [IEEE802154e] standard introduces some amendments to the IEEE 802.15.4 standard. Among its key features there is the Time-slotted Channel Hopping (TSCH), i.e., a novel MAC protocol, which better supports multi-hop communications in emerging industrial applications. In addition, it provides very few upgrades to security-related aspects.

Since the IEEE 802.15.4e amendment focuses only on link-layer aspects, the 6TiSCH WG was created to define open standards in support of the adoption of IPv6 over the TSCH mode of the IEEE802.15.4e standard, thus covering all facets related to the management of network communications in complex (and eventually distributed) Low-Power and Lossy Networks (LLNs) [I-D.ietf-6tisch-tsch] [I-D.wang-6tisch-6top].

Security aspects represent an important issue that needs to be considered in a 6TiSCH network. TSCH defines mechanisms to encrypt and authenticate MAC frames but it does not define how this keying material is generated [IEEE802154]. For this reason, the 6TiSCH WG needs to define (i) security requirements and related architecture,

(ii) join processes, (iii) the keying material and authentication mechanism needed by a new mote to join an existing network; (iv) a mechanism to allow for the secure transfer of application data between neighbor motes; and (v) define a mechanism to allow for the secure transfer of signaling data between motes and 6TiSCH.

The description of the security architecture and related architectural elements are being investigated in [I-D.draft-richardson-6tisch-security-architecture] and [I-D.draft-struik-6tisch-security-architecture-elements], respectively.

This Internet Draft focuses on layer-2 security aspects and describes standard compliant procedures for configuring layer-2 security services in IEEE 802.15.4e-based Low-power and Lossy Networks. In particular, main features covered by this document are:

- a review of security aspects presented in both IEEE 802.15.4 and IEEE 802.15.4e specifications, with particular attention to the set of parameters that need to be set for enabling security services at the MAC layer;

- the definition of types and properties of layer-2 keys;

- the classification of possible secure network configurations, which include Fully Secure, Unsecure, Partial Secure, and Hybrid Secure networks;

- the description of a set of consecutive steps (i.e., Setting-up, Bootstrap, Join, and Key Negotiation phases) that are required to establish a layer-2 secure link among a couple of nodes;

- the design of a lightweight Key Management Protocol useful for negotiating a per-peer layer-2 key.

3 Security in IEEE 802.15.4-2011 (and IEEE 802.15.4e-2012)

This section summarizes security features defined within IEEE 802.15.4 and IEEE 802.15.4e specifications [IEEE802154] [IEEE802154e].

The standard defines eight security levels to protect MAC frames, as summarized in Fig. 1 and imposes the adoption of the CCM* algorithm to perform encryption and description procedures (which requires a key of **128 bit).**

Security level	Security attribute	Data Integrity	Data Confidentiality
0	None	No	No
1	MIC-32	Yes	No
2	MIC-64	Yes	No
3	MIC-128	Yes	No
4	ENC	No	Yes
5	ENC-MIC-32	Yes	Yes
6	ENC-MIC-64	Yes	Yes
7	ENC-MIC-128	Yes	Yes

Figure 1. Security levels available for a IEEE 802.15.4 network.

At the MAC layer, encryption and decryption operations are implemented within the "outgoing frame security" and the "incoming frame security" procedures, respectively. They use a number of security attributes, summarized in what follows:

- macKeyTable: it is composed by a set of KeyDescriptor elements. A specific KeyDescriptor element is created for each key, composed by (see Tab. 61 of the IEEE 802.15.4 standard for more details [IEEE802154]):

  - The KeyIdLookupList, which is a list of KeyIdLookupDescriptor entries. A KeyIdLookupDescriptor is composed by a set of parameters (see Tab. 65 of the IEEE 802.15.4 standard for more details [IEEE802154]), i.e., KeyIdMode, KeySource, KeyIndex, DeviceAddMode, DevicePANId, and DeviceAddress, that are used to identify the key within the macKeyTable.

  - The DeviceDescriptorHandleList, which contains pointers to DeviceDescriptor elements stored within the macDeviceTable. It is used to identify which devices may use the key.

  - The KeyUsageList, which is a list of KeyUsageDescriptor elements. A KeyUsageDescriptor is composed by the FrameType and the CommandFrameIdentifies fields that indicate the

frame type with which the considered key may be used (see Tab. 62 of the IEEE 802.15.4 standard for more details [IEEE802154]).

- The Key.

- macDeviceTable: it is composed by a set of DeviceDescriptor elements, providing some information about remote devices which the node can establish secure communication with. A dedicated DeviceDescriptor element is associated to each remote device. It is composed by a number of fields, i.e., PANId, ShortAddress, ExtAddress, FrameCounter, and Extemp, which collect information related to a specific remote device (see Tab. 64 of the IEEE 802.15.4 standard for more details [IEEE802154]).

- macSecurityLevelTable: it is made by a set of SecurityLevelDescriptor elements, which store details about the security level required for each MAC frame type and subtype. Fields belonging to the SecurityLevelDescriptor data structure are: FrameType, ComamndFrameIdentifier, SecurityMinimum, DeviceOverrideSecurityMinimum, and AllowedSecurityLevels (see Tab. 63 of the IEEE 802.15.4 standard for more details [IEEE802154]).

- macFrameCounter: it is an integer value storing the outgoing frame counter for the considered device. Its length depends from the configured macFrameCounterMode (in TSCH-enabled networks it represents the ASN [IEEE802154e]).

- macAutoRequestSecurityLevel: it is an integer value providing the security level used for automatic data requests.

- macAutoRequestKeyIdMode: it is an integer value indicating the key identifier mode used for automatic data requests. It is not valid if the macAutoRequestSecurityLevel attribute is set to 0x00.

- macAutoRequestKeySource: it represents a short or extended IEEE 802.15.4 MAC address, indicating the originator of the key used for automatic data requests. This attribute is not valid if the macAutoRequestKeyIdMode element is not valid or set to 0x00.

- macAutoRequestKeyIndex: it is an integer value storing the index of the key used for automatic data requests. It is not valid if the macAutoRequestKeyIdMode attribute is not valid or set to 0x00.

- macDefaultKeySource: it is the extended IEEE 802.15.4 MAC address of the originator of the default key used for key identifier mode 0x01.

- macPANCoordExtendedAddress: it represents the extended address of the PAN coordinator.

- macPANCoordShortAddress: it represents the short address assigned to the PAN coordinator.

-macFrameCounterMode: it is an integer value describing the size of the frame counter (i.e., 0x04 corresponds to a frame size of 4 octets; 0x04 corresponds to a frame size of 5 octets).

During the outgoing security procedure, the high layer uses the KeyIdMode parameter to select a specific key in the macKeyTable to be used for protecting the MAC frame.

The KeyIdMode is set to 00, 01, 10, and 11 in the case the key can be implicitly derived by both sender and the receiver and it is not specified in the message, the key is explicitly determined by the KeyIndex parameter stored into the MAC header and the macDefaultKeySource, the key can be derived by considering KeyIndex and KeySource fields stored into the MAC header (with KeySource representing the short address of the device that has generated the key), and the key can be derived by considering KeyIndex and KeySource fields stored into the MAC header (with KeySource representing the IEEE extended address of the device that has generated the key), respectively.

Both IEEE 802.15.4 and IEEE 802.15.4e standards do not provide any guideline to create (and or negotiate) keys, as well as to configure the aforementioned security MAC attributes. They just delegate upper layers to orchestrate such aspects.

5 Definition of layer-2 keys

In line with [I-D.draft-richardson-6tisch-security-architecture], a "production network" may use two different layer-2 keys, that are "production network key" and "per-peer L2 key".

The "production network key" is a secret shared by all the authorized nodes. It can be obtained only if the node is able to correctly complete the join procedure, which offers authorization and authentication services.

The "per-peer L2 key" is, instead, negotiated only between a couple of nodes through a KMP strategy.

In addition, a new layer-2 key, namely "master L2 key", is defined. It represents an initial secret, which is shared among all the motes and

configured by the manufacturer or by the network administrator before the network deployment. Note that a mote can be subjected to any kind of tamper attacks. Without any further shrewdness, an attacker that may physically access to the mote could extract the "master L2 key", thus compromising the security of the whole 6TiSCH network. Hence, it is very important to ensure the protection to that tampering attacks by using specific software-based and/or hardware-based mechanisms [Walters07][Becher2006].

Each layer-2 key is used to protect a specific set of messages. In particular, the "master L2 key" is used for protecting enhanced beacon messages and data frames carrying messages exchanged during the join procedure; the "production network key" is used for protecting broadcast messages and MAC frames exchanged during the Key Negotiation Phase; the "per-peer L2 key" is used for encrypting and authenticating messages exchanged between two nodes at the MAC layer.

"master L2 key" and "production network key" should be identified within the network by setting KeyIdMode to 0x01 (for both of keys) and the KeyIndex to 1 and 2, respectively. Differently, the "per-peer L2 key" should be explicitly identified within the network. Hence, its KeyIdMode should be set to 0x03 and KeySource and KeyIndex parameters should be set according to the couple of nodes that negotiated the key (more details can be found in Sec. 6).

As it will better described in the following sections, the master L2 key" is stored within the device during the Setting-Up Phase and configured as one of security MAC attribute at the end of the Bootstrap Phase. The "production network key" is obtained and configured as one of security MAC attribute during the Join Phase. Finally, the "per-peer L2 key" is negotiated and configured at the MAC layer during the Key Negotiation Phase.

4 Security Configurations

Based on the status and the configuration of security services, a "production network" may fall within one of the following security configurations:

- Fully Secured network: all the devices forming the network are configured to fully support security services and they have already obtained (or negotiated) all the keys defined in the previous section. It represents the most secured configuration: all packets are encrypted and authenticated by using specific

keys, which depend from the message they carry. Nodes that do not support security capabilities (or that are not in posses of all the information to joining the network, such as key materials and encryption and decryption algorithms) are not allowed to join the network.

- Unsecured network: security services are not supported. Even if in possession of security capabilities, any pair of nodes is not allowed to establish a secured communication. Differently for the Fully Secured scheme, this is the lowest security level. Since the data encryption, the message integrity, and the peer authentication are not implemented, all the MAC frames are exchanged in clear. Hence, the setup and the maintaining of the network are described by the standard and no further upgrades are required.

- Partial Secured network: only the integrity of message is supported.

- Hybrid Secured network: a network falls in this configuration when there still are a group of devices that have not yet authenticated by the network (because they have not yet correctly completed the join procedure).

The standard imposes to specify, for each kind of MAC packet, minimum security levels that should be guaranteed. These restrictions must be detailed for each remote device. To this end, SecurityMinimum, DeviceOverrideSecurityMinimum, and AllowedSecurityLevels parameters are stored into the DeviceDescriptor element (see Sec. 3) to define the minimum security level (i.e., one of those reported in Fig.1), the possibility to override the minimum security level (i.e., DeviceOverrideSecurityMinimum is just a boolean flag), and the list of allowed security levels in the case the minimum one could be overridden, respectively.

Focusing the attention on "production network" that is not in a hybrid (i.e., dynamic) configuration, these parameters must be set as reported in Fig. 2.

Attribute	Secured Network Configurations		
	Unsecured	Fully	Partial
SecurityMinimum	0	from 5 to 7	from 1 to 4

DeviceOverride-SecurityMinimum	FALSE	FALSE	FALSE
AllowedSecuri-tyLevelsvels	0	from 5 to 7	from 1 to 4

Figure 2. Setting of security attributes of the DeviceDescriptor element in each defined secure network configuration.

The Unsecured network configuration does not support any security features. Hence, both minimum and allowable security levels are set to 0 for all the MAC frames and the possibility to override such constraints is disabled for all devices.

If the Fully Secured configuration is enabled, the minimum security level must be chosen in the range [5,7], thus allowing the possibility to support the encryption and the authentication of messages. The manufacturer must set the default value to 7; it can be updated by the network administrator. The minimum security level must not be overridden by any devices and, as a consequence, the field AllowedSecurityLevels should contain only one value, equal to the minimum security level.

If the Partial Secured configuration is enabled, the minimum security level must be chosen in the range [1,4], thus allowing the possibility to support the authentication of messages. The manufacturer must set the default value to 4; it can be updated by the network administrator. The minimum security level must not be overridden by any devices and, as a consequence, the field AllowedSecurityLevels should contain only one value, equal to the minimum security level.

6 Establishing a secured layer-2 link

A layer-2 secure link can be established through the execution of four consecutive phases: Setting-up, Bootstrapping, Join, and Key Negotiation (see Fig. 3).

The Setting-up Phase is used to store into the device all the secrets required to initialize a secured domain. The Bootstrap Phase, whose implementation is different for both PAN coordinator and the "join node", is used for initializing security MAC attributes. The Join Phase is handled by upper layers for offering authorization and authentication services and allows the device to receive at the end the NetworkKey. Finally, the Key Negotiation Phase handles the Key Management Protocol (KMP) and it is used to negotiate a layer-2 key between a couple of

nodes that are directly connected at the MAC layer.

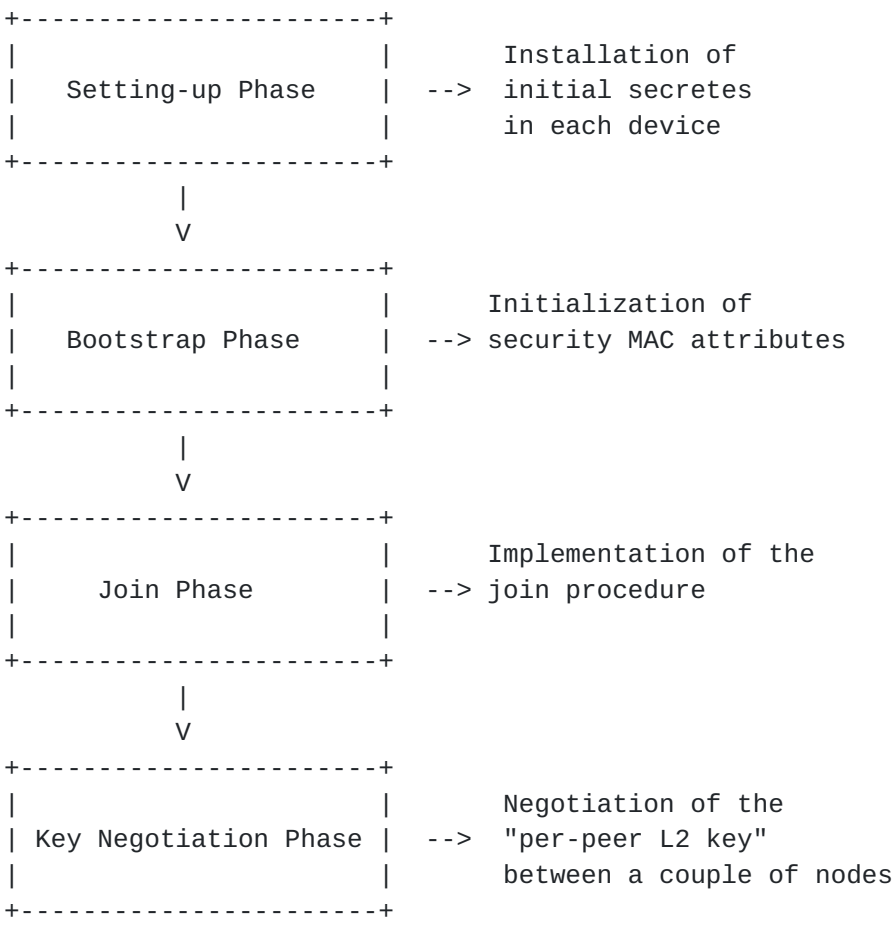

Figure 3. Summary of the proposed framework.

6.1 Setting-up phase

The setting-up phase is used to properly configure the device that will join to a "production network". It consists in storing, within the device, parameters and initial secrets, which will be used by secure algorithms and procedure to setup the secure domain. They include the "master L2 key", (ii) the GlobalSecurityLevelsTable, (iii) the private key of the node, (iv) the public key of the node stored within a certificate, and (iv) the certificate of the certification authority. This operation may be performed by the manufacturer or by the network administrator.

Note that the GlobalSecurityLevelsTable, that has been reported in Fig. 4, is used to store the minimum security level and the list of allowed security levels that must be adopted for each kind of MAC frame and for each security configuration defined in Sec. 5. Both the minimum security

level and the list of allowed security levels must be chosen by the manufacturer or by the network administrator, according to restrictions reported in Fig. 2.

Attribute	Frame Type	Secured Unsecured	Configurations Fully	Configurations Partial
Security Minimum	Beacon			
Security Minimum	Data			
Security Minimum	Command MAC			
Security Minimum	ACK			
AllowedSecurityLevels	Beacon			
AllowedSecurityLevels	Data			
AllowedSecurityLevels	Command MAC			
AllowedSecurityLevels	ACK			

Figure 4. Structure of the GlobalSecurityLevelsTable.

6.2 Bootstrap phase

6.2.1 Bootstrap phase for the PAN coordinator

As soon a node becomes the PAN coordinator, it should configure initial security MAC attributes, including those related to the "master L2 key". To this end, specific primitives of the 6top adaptation layer are used [I-D.wang-6tisch-6top].

The following operations are executed:

a) A CONFIGURE.security command is generated by the 6top layer and sent to the MAC entity to initialize security attributes. The set of parameters handled by this command are set as in the sequel:

a.1) enable = true;

a.2) macAutoRequestSecurityLevel = security level expected for the beacon message and stored within the GlobalSecurityLevelsTable;

a.3) macAutoRequestKeyIdMode = 0x03;

a.4) macAutoRequestKeySource = MAC address of the device;

a.5) macAutoRequestKeyIndex = 1;

a.6) macDefaultKeySource = MAC address of the device;

b) CONFIGURE.security.macSecurityLevelTable command is generated by the 6top layer and sent to the MAC entity to initialize macSecurityLevelTable. Parameters stored into this command are taken from the GlobalSecurityLevelsTable.

c) A new KeyIdLookupList data structure is created. A KeyIdLookupDescriptor is generated and stored into the KeyIdLookupList data structure. The KeyIdMode, the KeyIndex, and the key variables of this KeyIdLookupDescriptor are set to 0x01 and 1, respectively. Instead, KeySource, DeviceAddrMode, DevicePANId, and DeviceAddress are not set due to the selected KeyIdMode (see Tab. 65 of the IEEE 802.15.4 standard for more details [IEEE802154]).

d) A KeyUsageList data structure is created. One KeyUsageDescriptor for each kind of broadcast messages is create and stored into the KeyUsageList data structure.

e) An empty DeviceDescriptorHandleList is created. No data are stored within this list because the PAN coordinator does not yet know the list of devices that may use this key.

f) Then, the 6top layer deliver the "master L2 key", the KeyIdLookupList, the KeyUsageList, and the DeviceDescriptorHandleList to the MAC layer by using the CONFIGURE.security.macKeyTable primitive. Triggered by the CONFIGURE.security.macKeyTable command, the MAC layer will create a KeyDescriptor associated to the "master L2 key", where storing

all the parameters received by the 6top layer, and store it within the macKeyTable.

The Bootstrap phase for the PAN coordinator has been summarized in Fig. 5.

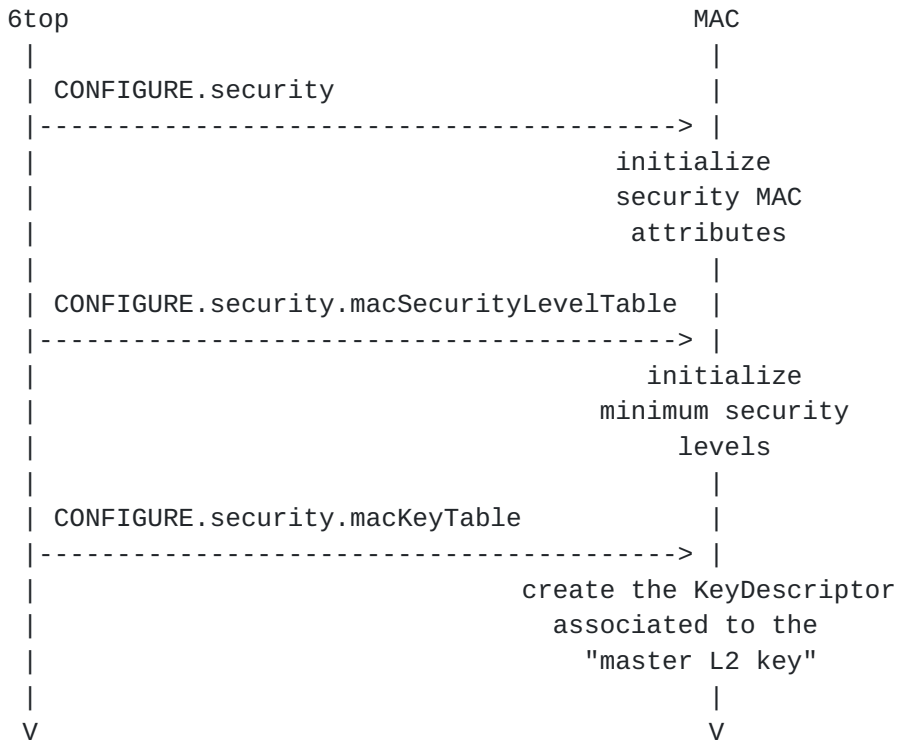

Figure 5. Bootstrap Phase for the PAN coordinator.

6.2.2 Bootstrap phase for a "joining node"

Before executing the Join Phase, a "joining node" should initialize security MAC attributes, including information related to the "master L2 key", through specific 6top adaptation layer primitives. To this end, after the reception of the enhanced beacon message, the following operations are executed:

a) From the received beacon message, the mote extracts the PAN_ID, the MAC address of the node that sent the beacon, and the FrameCounter.

b) A CONFIGURE.security primitive is generated by the 6top layer and sent to the MAC entity to initialize security attributes. The set of parameters handled by this primitive are set as in the sequel:

b.1) enable = true;

b.2) macAutoRequestSecurityLevel = security level expected for the beacon message and stored within the GlobalSecurityLevelsTable;

b.3) macAutoRequestKeyIdMode = 0x03;

b.4) macAutoRequestKeySource = MAC address of the device;

b.5) macAutoRequestKeyIndex = 1;

b.6) macDefaultKeySource = MAC address of the device;

c) CONFIGURE.security.macSecurityLevelTable primitive is generated by the 6top layer and sent to the MAC entity to initialize macSecurityLevelTable. Parameters stored into this command are taken from the GlobalSecurityLevelsTable.

d) A new KeyIdLookupList data structure is created. A KeyIdLookupDescriptor is generated and stored into the KeyIdLookupList data structure. The KeyIdMode and the KeyIndex variables of this KeyIdLookupDescriptor are set to 0x00, the MAC address of the node that sent the beacon message and 1, respectively. Instead, KeySource, DeviceAddrMode, DevicePANId, and DeviceAddress are not set due to the selected KeyIdMode (see Tab. 65 of the IEEE 802.15.4 standard for more details [IEEE802154]).

e) A KeyUsageList data structure is created. One KeyUsageDescriptor for each kind of messages is create and stored into the KeyUsageList data structure.

f) A new DeviceDescriptor element, associated to the node that sent the enhanced beacon message is created and stored into the macDeviceTable. It is built considering these specifications (see Tab. 64 of the IEEE 802.15.4 standard [IEEE802154] for more details):

f.1) The PANId variable is associated to the PAN_ID value extracted from the Beacon message.

f.2) The ShortAddress is set to the MAC address of node that sent the beacon message whenever the short addressing mode is used. This parameter is set to 0xfffe if only the extended addressing mode is used. If its value is unknown, the ShortAddress parameter is set to 0xfff.

f.3) The ExtAddress is set to the IEEE MAC address of node that sent the beacon message.

f.4) The FrameCounter parameter is set to the FrameCounter value extracted from the enhanced beacon message (it represents the ASN in the case the network works in TSCH-mode [IEEE802154e]).

f.5) The Exempt boolean flag is set to the allowed value of the DeviceOverriddeSecurityMinimum variable described in Fig. 2.

g) The DeviceDescriptorHandleList is created and populated with the DeviceDescriptor created at the previous step.

h) A KeyUsageList data structure is created and stored within the KeyDescriptor element. One KeyUsageDescriptor for each broadcast message is create and stored into the KeyUsageList data structure.

i) The 6top layer deliver the "master L2 key", the KeyIdLookupList, the KeyUsageList, and the DeviceDescriptorHandleList to the MAC layer by using the CONFIGURE.security.macKeyTable primitive. Triggered by the CONFIGURE.security.macKeyTable primitive, the MAC layer will create a KeyDescriptor associated to the "master L2 key", in which storing all the parameters received by the 6top layer, and will store it within the macKeyTable.

The Bootstrap Phase for a "joining node" has been summarized in Fig. 6.

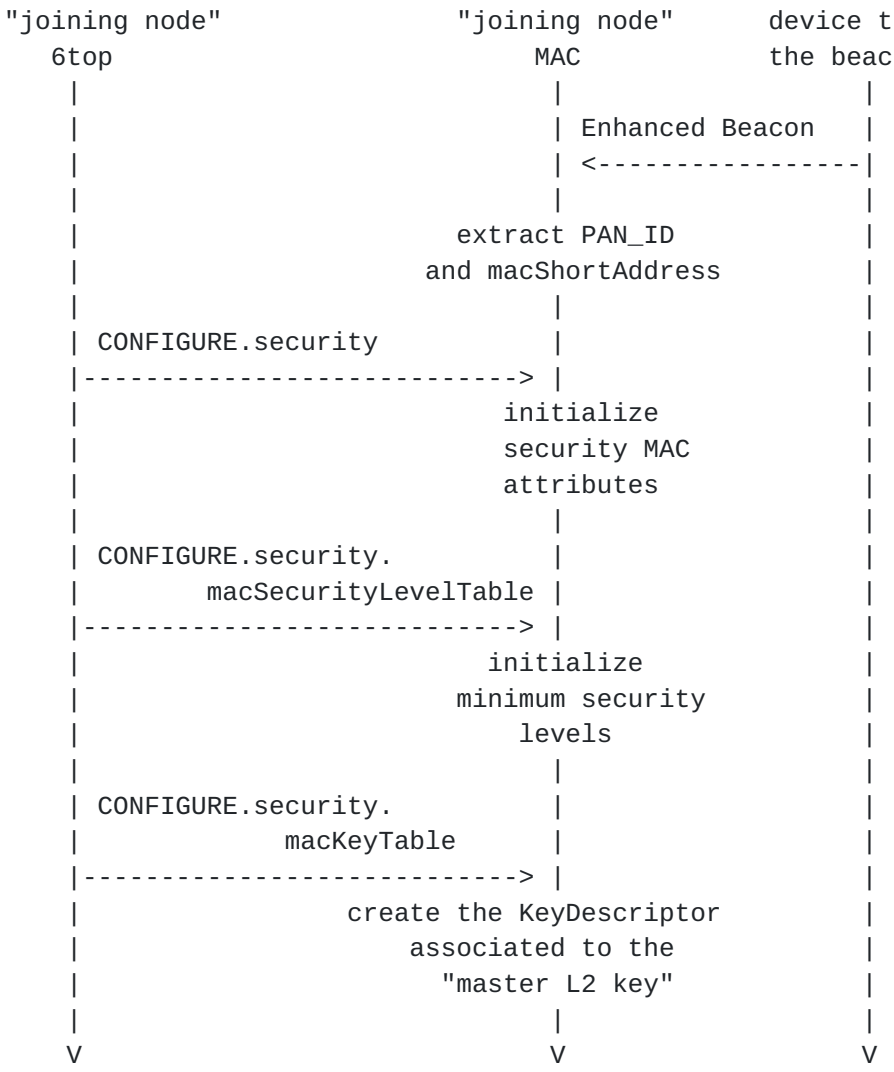

Figure 6. Bootstrap Phase for the "joining node".

6.3 Join Phase

During the Join Phase, the join procedure is implemented by upper layers for offering authorization and authentication features. This aspect is carefully investigated in both [I-D.draft-richardson-6tisch-security-architecture] and [I-D.draft-struik-6tisch-security-architecture-elements], thus being out of scope of this Internet Draft.

At the end of the join procedure, the "joining node" obtains the "production network key" and updates security MAC attributes as described for the "master L2 key", with the only difference that the KeyIndex is set to 2.

6.4 Key Negotiation Phase

Since resource-constrained devices are unable to perform complex algorithms and protocols [Altolini2013][Riaz2009], a simple key agreement protocol, based on both ECDH algorithm and Station-To-Station protocol [StsProtocol], is adopted during the execution of the key negotiation phase.

As described in Sec. 6.1, it is supposed that each node stores the certificate of the authority and the couple of its public and private key (generated through the adoption of elliptic curves). Obviously, the public key is stored within a certificate, signed by the authority.

In this section is described the KMP implemented between a couple of nodes, i.e., node A and node B, which want to negotiate a on-hop layer-2 key. Let PBK_A, PVK_A, PBK_B, and PVK_B be A's public key, A's private key, B's public key, and B's private key, respectively. Moreover, to handle the Key Negotiation phase, a number of high-level commands have been defined. In line with IEEE 802.15.4e specifications, they are mapped into specific Header Information Elements, each one identified by an unique element ID.

6.4.1 New Header Information Elements

The set of Header Information Elements introduced for handling the KMP are:

- Crypto Information Element (element ID set to 0x18). It is used to deliver the certificate storing the ECDH public key. Since the certificate length is generally higher than the IEEE 802.15.4e MAC payload, it is necessary to fragment the certificate, thus sending it through multiple consecutive MAC frames. To this end, the first byte of the introduced Information Element is used to indicate the fragment ID to which the current packet refers to. The second byte of the first fragment stores the RAND parameter, which is a random value adopted to finalize the mutual authentication.

- Authentication Information Element (element ID set to 0x19), which stored the AuthField used to execute the mutual authentication.

6.4.2 KMP description

The KMP consists of six consecutive steps:

- Step 1: node A sends to node B its certificate through a number of consecutive MAC frames containing the Crypto Information Element. Let RAND_A be the random number stored within the second byte of the Crypto Information Element belonging to the first MAC frame. All of these packets are protected by using the "production network key" received at the end of the Join Phase.

- Step 2: node B verifies the authenticity of the received certificate. In affirmative case, it sends to node A its certificate through a number of consecutive MAC frames containing the Crypto Information Element. Let RAND_B be the random number stored within the second byte of the Crypto Information Element belonging to the first MAC frame. All of these packets are protected by using the NetworkKey received at the end of the Join Phase.

- Step 3: node A and node B computes the PreLinkKey, P_k, by using the ECDH algorithm.

- Step 4: node A computes the authentication parameter as expected for the Station-To-Station protocol:

AuthField_A = E(P_k, sign),

where

sign = S(PVK_A, H_128 {P_k || RAND_B || RAND_A})

Then, it creates a Authentication Information Element containing the aforecomputed AuthField and sends it to node B. Note that H_128 {.}, E(.), and S(.) operators refer to a 128-bit hash function, the encryption, and the digital sign algorithm, respectively.

- Step 5: node B computes the authentication parameter through the 128-bit hash function, as in the sequel:

AuthField_B = E(P_k, sign),

where

sign = S(PVK_B, H_128 {P_k || RAND_A || RAND_B}).

Then, it creates a Authentication Information Element containing the aforecomputed AuthField and sends it to node A.

- Step 6: nodes A and B verifies the authenticity of received AuthField parameters (according to the Station-To-Station) protocol and computes the "per-peer L2 key".

6.4.2 Calculation of the "per-peer L2 key"

The standard imposes to use the CCM* algorithm and a 128-bit key to protect MAC frames. At the same time, the CCM* algorithm assumes that each key must be used for a specific number of block ciphers [IEEE802154].

For each i-th group of block ciphers, the "per-peer L2 key", L_k, is computed as in the following:

L_k = H_128 (i | PAN_ID | P_k).

Node A and node B compute the "per-peer L2 key" and updates mac security attributes accordingly. To this end, the following steps are executed:

a) If i=1, a new DeviceDescriptor element, associated to the remote mote with which it has negotiated the "per-peer L2 key", is created. It is composed of:

a.1) the PANId, which is set to the PAN_ID value.

a.2) The ShortAddress, which is set to the MAC address of the remote node whenever the short addressing mode is used. This parameter is set to 0xfffe if only the extended addressing mode is used. In the case its value is unknown, this parameter is set to 0xfff.

a.3) The ExtAddress, which is set to the IEEE MAC address of the remote node.

a.4) The FrameCounter, which is set to the FrameCounter value extracted from the latest packet received by the remote node.

a.5) The Exempt boolean flag, which is set to the allowed value of the DeviceOverriddeSecurityMinimum variable described in Fig. 2.

b) A new KeyIdLookupList data structure is created. A KeyIdLookupDescriptor is generated and stored into the KeyIdLookupList data structure. The KeyIdMode, the KeySource, and the KeyIndex variables of this KeyIdLookupDescriptor are set to 0x03, the MAC address of the remote mote, and 1, respectively. DeviceAddrMode, DevicePANId, and DeviceAddress are not set because of the selected KeyIdMode (see Tab. 65 of the IEEE 802.15.4 standard for more details [IEEE802154]).

c) A KeyUsageList data structure is created and stored within the KeyDescriptor element. One KeyUsageDescriptor associated to data MAC frames is created and stored into the KeyUsageList data structure.

d) A DeviceDescriptorHandleList is created and populated with the pointer to the DeviceDescriptor created at the point a).

e) The 6top layer delivers the "per-peer L2 key", the KeyIdLookupList, the KeyUsageList, and the DeviceDescriptorHandleList to the MAC layer by using the CONFIGURE.security.macKeyTable command. Triggered by the CONFIGURE.security.macKeyTable command, the MAC layer will create a KeyDescriptor associated to the "per-peer L2 key", L_k, in which storing all the parameters received by the 6top layer, and will store it within the macKeyTable.

7 Security Considerations

There are no security considerations for this document.

8 IANA Considerations

There is no IANA action required for this document.

9 References

9.1 Normative References

[IEEE802154] IEEE standard for Information Technology, "IEEE std. 802.15.4, Part. 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks", June 2011.

[IEEE802154e] IEEE standard for Information Technology, "IEEE std. 802.15.4e, Part. 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendament 1: MAC sublayer", April 2012.

[I-D.ietf-6tisch-tsch] Watteyne, T., MR. Palattella, and LA. Grieco, "Using IEEE802.15.4e TSCH in an LLN context: Overview, Problem Statement and Goals", Internet-Draft draft-ietf-6tisch-tsch-03, October 2014.

[I-D.wang-6tisch-6top] Wang, Q., Vilajosana, X. and T. Watteyne, "6TiSCH Operation Sublayer (6top)", Internet-Draft draft-wang-6tisch-6top-sublayer-01, July 2014.

[I-D.ietf-6tisch-terminology] Palattella, MR., Ed., Thubert, P., Watteyne, T., and Q. Wang, "Terminology in IPv6 over Time Slotted Channel Hopping". Internet Draft draft-ietf-6tisch-terminology-02, July 2014.

[I-D.draft-richardson-6tisch-security-architecture] M. Richardson, "security architecture for 6top: requirements and structure". Internet Draft draft-richardson-6tisch-security-architecture-02 April 2014.

[I-D.draft-struik-6tisch-security-architecture-elements] R. Struik, Y. Ohba, and S. Das, "6TiSCH Security Architectural Elements, Desired Protocol Properties, and Framework". Internet Draft draft-struik-6tisch-security-architecture-elements-01 October 2014.

[DH] W. Diffie and M. Hellman, "New directions in cryptography," IEEE Trans. Inf. Theor. 22, 6 Sep., 2006.

[StsProtocol] Whitfield Diffie, Paul C. van Oorschot and Michael J, "Wiener, Authentication and authenticated key exchange", Designs, Codes, and Cryptography, 1987.

9.2 Informative References

[ZIGBEEIP] ZigBee Public Document 15-002r00, "ZigBee IP Specification", 2013.

[Camtepe2005] Seyit A. Camtepe and Bulent Yener, "Key Distribution Mechanisms for Wireless Sensor Networks: a Survey", Technical Report 2005.

[Walters07] John Paul Walters, Zhengqiang Liang, Weisong Shi, and Vipin Chaudhary, "Wireless sensor network security: A survey," in book chapter of Security", Proc. of Distributed, Grid, and Pervasive Computing, CRC Press, 2007.

[Wang2006] Yong Wang, Garhan Attebury, and Byrav Ramamurthy, "A survey of security issues in wireless sensor networks", IEEE Communications Surveys & Tutorials, 2006

[Cayirci2007] Security in Wireless Ad Hoc and Sensor Networks. John Wiley & Sons, 2007.

[RFC5191] Forsberg, D., Ohba, Y., Patil, B., Tschofenig, H., and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA)", RFC 5191, May 2008.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, January 2012.

[HIPDEX] Moskowitz, R., "HIP Diet EXchange (DEX)", draft-moskowitzhip-rg-dex-06 (work in progress), May 2012.

[PalattellaSurvey] Maria Rita Palattella, Nicola Accettura, Xavier Vilajosana, Thomas Watteyne, Luigi Alfredo Grieco, Gennaro Boggia, and Mischa Dohler," Standardized Protocol Stack For The Internet Of (Important) Things", IEEE Communications Surveys & Tutorials, December, 2012

[StallingsSecurityBooks] William Stallings: Cryptography and network

security - principles and practice. Prentice Hall 2010.

[Becher2006] Alexander Becher, Zinaida Benenson, and Maximillian Dornseif, "Tampering with motes: real-world physical attacks on wireless sensor networks", In Proc. of conf. on Security in Pervasive Computing (SPC), Berlin, 2006

[TELOSB] "Crossbow Technology, TelosB Datasheet." [Online]. Available: http://www.willow.co.uk/TelosB_Datasheet.pdf

[Riaz2009] Riaz, R.; Ki-Hyung Kim; Ahmed, H.F., "Security analysis survey and framework design for IP connected LoWPANs," Autonomous Decentralized Systems, 2009. ISADS '09. International Symposium on , vol., no., pp.1,6, 23-25 March 2009

[Altolini2013] Altolini, D.; Lakkundi, V.; Bui, N.; Tapparello, C.; Rossi, M., "Low power link layer security for IoT: Implementation and performance analysis," Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International , vol., no., pp.919,925, 1-5 July 2013

[Watteyne2012] Thomas Watteyne, Xavier Vilajosana, Branko Kerkez, Fabien Chraim, Kevin Weekly, Qin Wang, Steven D. Glaser, Kris Pister: OpenWSN: a standards-based low-power wireless development environment. Trans. Emerging Telecommunications Technologies 23(5): 480-493 (2012)

Authors' Addresses

Giuseppe Piro
DEI, Dep. of Electrical and Information Engineering
Politecnico di Bari
Via Orabona 4, 70125, Bari, ITALY
Phone: +39 0805963301

Email: giuseppe.piro@poliba.it

Gennaro Boggia
DEI, Dep. of Electrical and Information Engineering

Politecnico di Bari
Via Orabona 4, 70125, Bari, ITALY
Phone: +39 0805963913

Email: gennaro.boggia@poliba.it

Luigi Alfredo Grieco
DEI, Dep. of Electrical and Information Engineering
Politecnico di Bari
Via Orabona 4, 70125, Bari, ITALY
Phone: +39 0805963911

Email: alfredo.grieco@poliba.it