6TiSCH
INTERNET-DRAFT
Intended Status: Informational
Expires: April 21, 2014

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

October 18, 2013

A standard compliant security framework for Low-power and Lossy Networks

draft-piro-6tisch-security-issues-00

Abstract

The aim of this Internet Draft is to define a standard compliant security framework for Low-power and Lossy Networks, in order to enable the possibility to encrypt and authenticate messages exchanged by nodes at the MAC layer. The framework is fully compatible with both IEEE 802.15.4 and IEEE 802.15.4e standards and offers a wide range of security features to network architectures developed within the 6TiSCH Working Group. In particular, it offers: (i) different kinds of security architectures; (ii) an efficient mechanism for initializing a secure IEEE 802.15.4 domain; and (iii) a lightweight mechanism to negotiate link keys among devices.

Status of this Memo

Table of Contents

1 Acronyms

The following acronyms are used in this document:

DH Diffie Hellman

DEX HIP Diet EXchange

DTLS Datagram Transport Layer

LLN Low-power and Lossy Network

LBR LLN Border Routers

FFD Full Function Device

HIP Host Identity Protocol

MAC Medium Access Control

PAN Personal Area Network

PANA Protocol for Carrying Authentication for Network Access

PHY Physical

RFD Reduced Function Device

RSA Rivest-Shamir-Adleman

TSCH Time-slotted Channel Hopping

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 (PHY) layers of the protocol stack and supports the possibility to protect MAC packets by means of symmetric-key cryptography techniques with several security options.

This standard relies on upper layers to orchestrate, enable, configure, and negotiate security services, as well as to handle the creation and exchange of encryption keys. But it leaves open some aspects, such as the initialization of a secure IEEE 802.15.4 domain, the generation and the exchange of keys, the management of joining operations in a secure 802.15.4 network already configured in the

past.

The IEEE 802.15.4e standard introduces some amendments to the IEEE 802.15.4 standard. The most important one is the Time-slotted Channel Hopping (TSCH), i.e., a novel MAC protocol, which better supports multi-hop communications in emerging industrial applications.

Since the IEEE 802.15.4e amendment focuses only on link-layer aspects, the 6TiSCH Working Group was born to suggest the adoption of IPv6 over the TSCH, thus covering all facets related to the schedule of network communications in complex (and eventually distributed) Low-Power and Lossy Networks (LLNs).

In industrial environments, security issues are very important. Hence, a complete framework, which supports a wide range of security features, is highly required.

In this context, the security in LLNs has been firstly addressed within ZigBee IP specifications, i.e., a suite of high level communication protocols sitting on top of the IEEE 802.15.4 MAC [ZIGBEEIP]. ZigBee IP supports end-to-end and link-layer security and a public key infrastructure based on X.509 certificates. It imposes the adoption of the same link-key (shared among all nodes) to protect packets belonging to any kind of services (i.e., only a single security level is allowed). This makes the network highly sensible to the presence of compromised devices. Moreover, the cost needed to update the key within the network could be high, especially for heavy loaded systems.

Security aspects have been also faced in [6TSCH-SEC] and [GARCIA-SEC-IOT]. The work [6TSCH-SEC] introduces a simple security framework based on four consecutive phases (i.e., implanting, bootstrapping, link establishment, and operational phases) which allow the initialization of a secured IEEE 802.15.4 network. It exploits well-known approaches, like PANA [RFC5191], HIP DEX [HIPDEX], and DTLS [RFC6347], to authenticate nodes and to negotiate link keys. Instead, [GARCIA-SEC-IOT] defines a set of security profiles to guarantee in different scenarios (e.g., network without security requirements, home, managed home, industrial, and advanced industrial). For each scenario, it identifies a number of security threats and suggests how fixing them through well-known protocols and algorithms.

Proposals presented in [6TSCH-SEC] and [GARCIA-SEC-IOT] do not provide a complete definition of a security framework for LLNs, which offers, at the same time, an accurate procedure for the network initialization, the support for different security configurations, and a fully compatibility with the standard. In addition, they do not guarantee the real possibility to implement conceived proposals in

devices, i.e., sensors with very limited computational, energy, and storage capabilities.

This Internet Draft proposes a complete and standard compliant framework offering a number of security features at the MAC layer of LLNs. All aspects have been designed in order to ensure an easy and effectively implementation on real devices.

In particular, the security framework described in this document introduces all the details required to enable the security for both IEEE 802.15.4 and IEEE 802.15.4e networks, as well as the possibility to encrypt and authenticate any kind of communications devised by the 6TiSCH Working Group.

In summary, the security framework covers: (1) five different kinds of security configurations; (2) an efficient mechanism for initializing a secure IEEE 802.15.4 (or IEEE 802.15.4e) domain; (3) a lightweight mechanism to negotiate link keys among devices; (4) a very simple technique adopted to update link keys during the time.

3 Security in IEEE 802.15.4

This section summarizes security features defined within the standard [IEEE802154], thus simplifying the comprehension of the remaining part of this Internet Draft.

The IEEE 802.15.4 standard defines eight security levels to protect MAC frames, as summarized in Fig. 1.

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 functionalities are implemented within the "outgoing frame security" and the "incoming frame security" procedures, respectively. They exploits 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 a 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.

- 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.

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 derived implicitly by both sender and the receiver and its is not specified in the message, the key is determined explicitly 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.

The IEEE 802.15.4 standard does not provide any guideline to create (and or negotiate) keys, as well as to configure the aforementioned security MAC attributes.

4 Definition of the secured domain

In this document, the "secured domain" concept refers to the portion of a LLN network where procedures and techniques described in this proposal must be performed in order to configure and maintain secured communications.

Two types of network nodes, i.e., the Full Function Device (FFD) and the Reduced Function Device (RFD), can be found in an IEEE 802.15.4 network. They can be arranged in both peer-to-peer and star topologies. A FFD device has the highest computational capabilities and it works as the coordinator of the network (also called PAN coordinator), thus being a reference node for a group of other RFD devices. Instead, a RFD device has lower resource and communication capabilities and it is able to communicate only with its reference FFD. Whereas RFD nodes must be connected only to coordinator, FFD devices can connect among them forming more complex meshed network architectures.

The concept of secured domain adopted in this Internet Draft coincides with the broadcast domain of an IEEE 802.15.4 network, which is composed by a number of RFD connected to a coordinator and the coordinator itself.

The IEEE 802.15.4e amendment does not introduces any variations to the network architecture, thus the previous definition is still valid in this context.

The same consideration can be done for network architectures and models designed within the 6TiSCH Working Group that, as explained before, are based on top of the IEEE 802.15.4e amendment.

Definitively, to the sake of clarity, from this moment on, we will focus on the terminology related to the IEEE 802.14.5 standard, implying that the entire framework can be adopted also for IEEE 802.15.4e networks and 6TiSCH's proposals.

5 Security Configurations

The security framework described in this document supports five network configurations, which differ among them according to the offered security features. They are:

- Fully Secured network: all the IEEE 802.15.4 devices forming the network are configured to fully support security services. It represents the most secured configuration: all packets, independently from the message they carry, are encrypted and authenticated. 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 offers 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: the network can be composed by heterogeneous nodes that could or could not support security features. As default, the network is created in an unsecured manner. All the non-unicast control messages sent by the coordinator should be transmitted in clear. In this way, in fact, it is ensured that all the devices are able to read the content of packets. A RFD node with security capabilities, that intends to exchange encrypted and/or authenticated packets with the coordinator, could negotiate a set of link key with its reference FFD.

- Flexible Secured network: as default, the network is setup with the Fully Secured configuration and all packets are encrypted and authenticated. If there is at least one node that have not security capabilities, the coordinator could decides to switch to the Hybrid Secured configuration.

5.1 Minimum security requirements

The IEEE 802.15.4 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.

With reference to secured network configurations presented in Sec. 3.1, these parameters must be set as reported in Fig. 2.

Attribute	Secured Network Configurations				
	Unsecured	Fully	Partial	Hybrid	Flexible
SecurityMinimum	0	from 5 to 7	from 1 to 4	0	from 1 to 7
DeviceOverride-SecurityMinimum	FALSE	FALSE	FALSE	FALSE	TRUE
AllowedSecuri- tyLevelsvels	0	from 5 to 7	from 1 to 4	from 0 to 7	from 0 to 7

Figure 2. Setting of security attributes of the DeviceDescriptor element in each proposed 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.

If the Hybrid Secured configuration is enabled, the minimum security level must be set to 0, thus supporting the joining of devices having different security capabilities. All the security levels could be allowed and the network administrator could decide to enable only a subset of them according to the network design.

If the Flexible Secured configuration is enabled, the minimum security level must be set to 1. The joining of nodes without (or with limited) security capabilities is permitted by setting the DeviceOverrideSecurityMinimum variable to TRUE and by including lower security levels in the list of AllowedSecurityLevels.

6 Initialization of a secured IEEE 802.15.4 domain

A secured 802.15.4 domain is created by means of three different phases: setting-up, bootstrapping, and key negotiation.

6.1 Setting-up phase

The setting-up phase is used to properly configure the device that will operate in an IEEE 802.15.4 network.

It consists in storing, within the device, parameters and initial secrets (i.e., the masterKey), which will be used by secure algorithms and procedure to setup the secure domain. They include. (i) the MasterKey, (ii) the PrimeNumbersTable, and (iii) the GlobalSecurityLevelsTable.

This operation may be performed by the manufacturer or by the network administrator. The MasterKey can be used to generate two different keys:

- the DefaultKey, exploited to protect broadcast messages (i.e., the beacon frame);

- the LinkKey, used to encrypt and authenticate unicast packets (i.e., those exchanged between only two specific nodes).

The GlobalSecurityLevelsTable, that has been reported in Fig. 3, 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. The table reported in Fig. 3 does not consider ACK messages because they must not be protected (as imposed by the IEEE 802.15.4 standard [IEEE802154]).

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 Network Configurations			
		Fully	Partial	Hybrid	Flexible
Security Minimum	Beacon				
Security Minimum	Data				
Security Minimum	Command MAC				
AllowedSecurityLevels	Beacon				
AllowedSecurityLevels	Data				
AllowedSecurityLevels	Command MAC				

Figure 3. Structure of the GlobalSecurityLevelsTable.

The PrimeNumbersTable stores a set of prime numbers and their respective primitive roots, which are used during the key negotiation procedure. Its implementation is reported in Fig. 4.

PrimeNumberId	Prime Number	Primitive Root
1	p1	g1
2	p2	g2
:		:
N	pN	gN

+----------------+-------------+----------------+
Figure 4. Structure of the PimeNumbersTable.

6.2 Bootstrap phase for a FFD device

The PAN is initialized by a FFD node. The MAC entity of the FFD node starts scanning the channel (for discovering the presence of other active coordinators and identifying the portion of the spectrum which it could operate in) after the reception of the MLME-START.request primitive, generated by the so called Next Higher Layer. Then, it will answer with a MLME-START.confirm primitive reporting a SUCCESS status. From this moment on, the device behaves as the PAN coordinator and it is able to select the identification number for the PAN, i.e., the PAN_ID, and the short MAC address of the IEEE 802.15.4 network [IEEE802154]. In this phase, the FFD generates the DefaultKey, D_k, and updates MAC security attributes accordingly. The DefaultKey, D_k, will be used to encrypt/decrypt all the broadcast messages. To this end, the following tasks will be executed.

a) The DefaultKey, D_k, is obtained from the MasterKey, M_k, by using a 128-bit hash function, H_128(.)

D_k= H_128(PAN_ID | M_k).

b) The FFD creates the KeyDescriptor associated to the DefaultKey, D_k, by following these steps:

b.1) A new KeyIdLookupList data structure is created and stored within the KeyDescriptor element. 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 device, and 1, respectively. Instead, 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]).

b.2) 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.

b.3) The DeviceDescriptorHandleList is left blank because the FFD does not yet know the list of devices that may use

this key.

b.4) The DefaultKey, D_k, is stored within the Key field.

c) The KeyDescriptor created in the previous step is added to the macKeyTable.

d) The macDefaultKeySource is set to the MAC address of the device.

6.3 Bootstrap phase for a RFD device in a Beacon-enabled network

To join the network, the RFD device should associate with the coordinator. The Next Higher Layer sends to the MAC entity the MLME-ASSOCIATE.request primitive, starting the association procedure.

In this phase, the FFD generates the DefaultKey, D_k, and updates MAC security attributes accordingly. The DefaultKey, D_k, will be used to encrypt/decrypt all the broadcast messages. To this end, the following tasks will be executed:

a) The RFD node receives a Beacon messages sent by the coordinator and extracts from it the PAN_ID, the MAC address of the coordinator and the FrameCounter of the received frame.

b) A new DeviceDescriptor element, associated to the coordinator (i.e., the FFD node that sent the 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):

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

b.2) The ShortAddress is set to the MAC address of the coordinator 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.

b.3) The ExtAddress is set to the IEEE MAC address of the coordinator.

b.4) The FrameCounter parameter is set to the FrameCounter value extracted from the Beacon message.

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

c) The DefaultKey, D_k, is obtained from the MasterKey, M_k, by using a 128-bit hash function, H_128(.):

D_k= H_128(PAN_ID | M_k).

d) The RFD creates the keyDescriptor associated to the DefaultKey, D_k, by following these steps:

d.1) a new KeyIdLookupList data structure is created and stored within the KeyDescriptor element. 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 coordinator that sent the Beacon frame, and 1, respectively. Instead, 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.2) 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.

d.3) The DeviceDescriptorHandleList is created and populated with the pointer to the DeviceDescriptor created at the point 2.

d.4) The DefaultKey, D_k, is stored within the Key field.

e) The KeyDescriptor created in the previous step is added to the macKeyTable.

f) The macDefaultKeySource is set to the MAC address of the coordinator.

6.4 Bootstrap phase for a RFD device in a not-Beacon-enabled network

In the case the not-beacon-enabled scheme is enabled, the RFD device must explicitly request its generation to the coordinator. The payload of the Beacon Request packet must be protected using an ephemeral key, phi_k, obtained from the MasterKey, M_k, and the source address of the

RFD node, srcMACaddress, as

phi_k = H_128(srcMACaddress | M_K).

The KeyIdMode of the Beacon Request packet is set to 00, thus enabling the coordinator to implicitly obtain the ephemeral key.

Once received the Beacon frame, the RFD node will execute all the steps described in Sec. 6.3.

6.5 Key negotiation phase Since resource-constrained devices are unable to perform complex algorithms and protocols, a simple key agreement protocol is adopted during the execution of the key negotiation phase.

A new command message, which is identified with a CommandFrameIdentifier set to 0xAA, is used for this purpose. It is composed by four different fields: KeyGenControlField, Rand, KeyMaterial, and AuthenticationField.

The structure of the new command MAC frame has been reported in Fig. 5. The structure of the KeyGenControlField, instead, are shown in Fig. 6. The introduction of these new fields respects the constraints imposed by the standard about the maximum packet size.

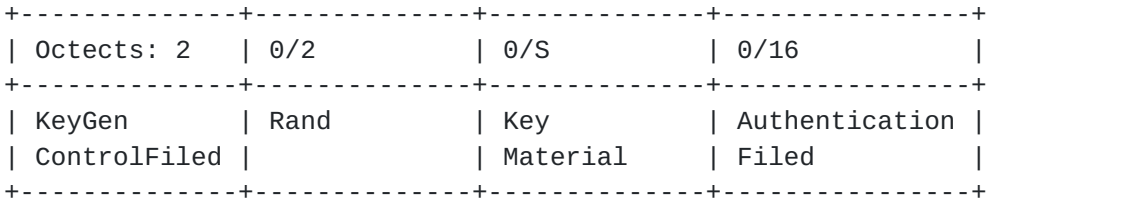

Figure 5. A new command MAC frame adopted during the key negotiation phase.

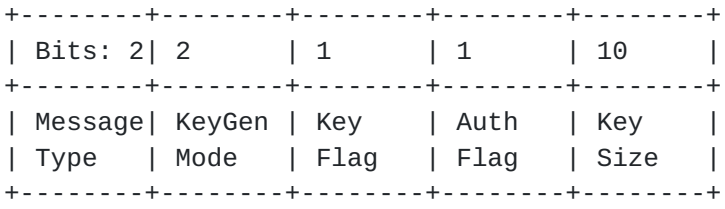

Figure 6. KeyGenControlField of the new command MAC frame adopted during the key negotiation phase.

The KeyGenControlField (2 bytes long) stores details about the content of the message. It is composed by the following fields:

- the MessageType (2 bits long), which identifies the type of message exchanged during the procedure. It may assume these

values:

- MessageType=00 identifies a message storing key materials (i.e., DH parameters).

- MessageType=01 identifies a message storing key materials belonging to a different approach selected before by the remote node. This message can be generate only by the FFD in the case the key negotiation algorithm chosen by the RFD is not supported.

- MessageType=10 identifies final messages belonging to the key negotiation phase that are used to verify the mutual authentication of nodes.

- MessageType=11 is reserved for future upgrades.

- the KeyGenMode (2 bits long), which describes the algorithm adopted for key generation. In this Internet draft we describe a key negotiation procedure based on the DH algorithm, which is identified by the code 00. Other values, i.e., 01, 10 and 11, are reserved and can be used for future upgrades.

- the boolean KeyFlag (1 bit long), which is set to TRUE in the case the message delivers key materials or to FALSE otherwise.

- the boolean AuthFlag (1 bit long)), which is set to TRUE in the case the message delivers an authentication field or to FALSE otherwise.

- the KeySize (10 bits long), which indicates the size of the transported key material. Its value is set to 0 in the case the message does not contain any key materials.

The Rand field (0/2 bytes long) contains a random value used for generating the PreLinkKey, P_k, and for verifying the authenticity of the remote device. It is present only if MessageType is equal to 00 or 01.

The KeyMaterial field (0/S bytes long, where S is the size of the prime number) contains key materials, such as DH parameters. It is present only if MessageType is equal to 00 or 01.

AuthenticationField field (0/16 bytes long) is used to verify the authenticity of the remote device. It is present only if MessageType is equal to 10.

6.5.1 Key exchange mechanism based on DH

The key exchange mechanism based on the DH algorithm is reported in Fig. 7.

It is initialized by the RFD device that wants to establish a secure link with the remote FFD. The procedure assumes that both RFD and FFD devices store into the PrimeNumbersTable the same set of N prime numbers and their primitive roots, each one having size equal to S (see Sec. 6.1 for more details).

The number of bits needed to identify each prime number of the PrimeNumbersTable, i.e., P_bits, is equal to

P_bits = log2 (N).

The key exchange mechanism is provides the execution of these operations:

a) The RFD identifies in the PrimeNumbersTable a prime number, P, and the corresponding primitive root, g, by extracting the latest P_bits bits from the output of the following hash function

H_128(PAN_ID | D_k).

b) The RFD computes the private key and the public key, according to the DH algorithm. Hence, the private key, PVK_RFD, is extracted as a random number. Then, the public key, PBK_RFD, is created as:

PBK_RFD = g^PVK_RFD * mod P

c) The RFD extract another random number, RAND_1, that will be used for the mutual authentication.

d) The RDF sends its public key, PBK_RFD, to the remote FFD through a specific MAC command frame, which is composed by the following fields (see Fig. 7): MessageType=00, KeyGenMode=01, KeyFalg=TRUE, AuthFlag=FALSE, KeySize=S, Rand=RAND_1, and KeyMaterial=PBK_RFD. This message is encrypted with the DefaultKey, D_k.

e) Once received the aforementioned MAC command frame, the FFD node generates, in turn, its private and public key through the DH algorithm. Hence, it identifies in the PrimeNumbersTable a prime number, P, and the corresponding primitive root, g, by extracting the latest P_bits bits from the output of the following hash function

H_128 (PAN_ID | D_k).

Then, it generates a random variable, PVK_FFD, which is the private key and computes the public key, PBK_FFD, as in the following:

PBK_FFD = g^PVK_FFD * mod P.

f) The FFD extract another random number, RAND_2, that will be used for the mutual authentication;

f) The RDF sends its public key, PBK_FFD, to the remote RFD through a specific MAC command frame, which is composed by the following fields (see Fig. 7): MessageType=00, KeyGenMode=01, KeyFalg=TRUE, AuthFlag=FALSE, KeySize=S, Rand=RAND_2, and KeyMaterial=PBK_FFD. This message is encrypted with the DefaultKey, D_k.

h) The RFD computes the PreLinkKey, P_k

P_k = PBK_FFD^PVK_RDF * mod P.

i) The FFD computes the PreLinkKey, P_k,

P_k = PBK_RFD^PVK_FFD * mod P.

j) Both RFD and FFD devices computes the LinkKey by using the procedure described in Sec. 6.5.2 and update MAC security attributes according to procedures described in Secs. 6.5.3 and 6.5.4.

k) The RFD node computes the authentication parameters, AUTH_RFD, through the 128-bit hash function, H_128(.),

AUTH_RFD=H_128(P_k || RAND_2 || RAND_1).

Then, it sends to the coordinator a new MAC command message to demonstrate its authenticity. This message is composed by the following fields (see Fig. 7): MessageType=10, KeyGenMode=01, KeyFalg=FALSE, AuthFlag=TRUE, KeySize=0, Rand=RAND_2, and AuthenticationField=AUTH_RFD. This message is protected by using the LinkKey computed before.

l) The FFD node computes the authentication parameters, AUTH_FFD, through the 128-bit hash function, H_128(.)

AUTH_FFD=H_128 (P_k || RAND_1|| RAND_2).

Then, it sends to the RFD node a new MAC command message to demonstrate its authenticity. This message is composed by the following fields(see Fig. 7): MessageType=10, KeyGenMode=01, KeyFalg=FALSE, AuthFlag=TRUE, KeySize=0, Rand=RAND_2, and AuthenticationField=AUTH_FFD. This message is protected through the LinkKey computed before.

m) Both RFD and FFD verify the authenticity of the remote.

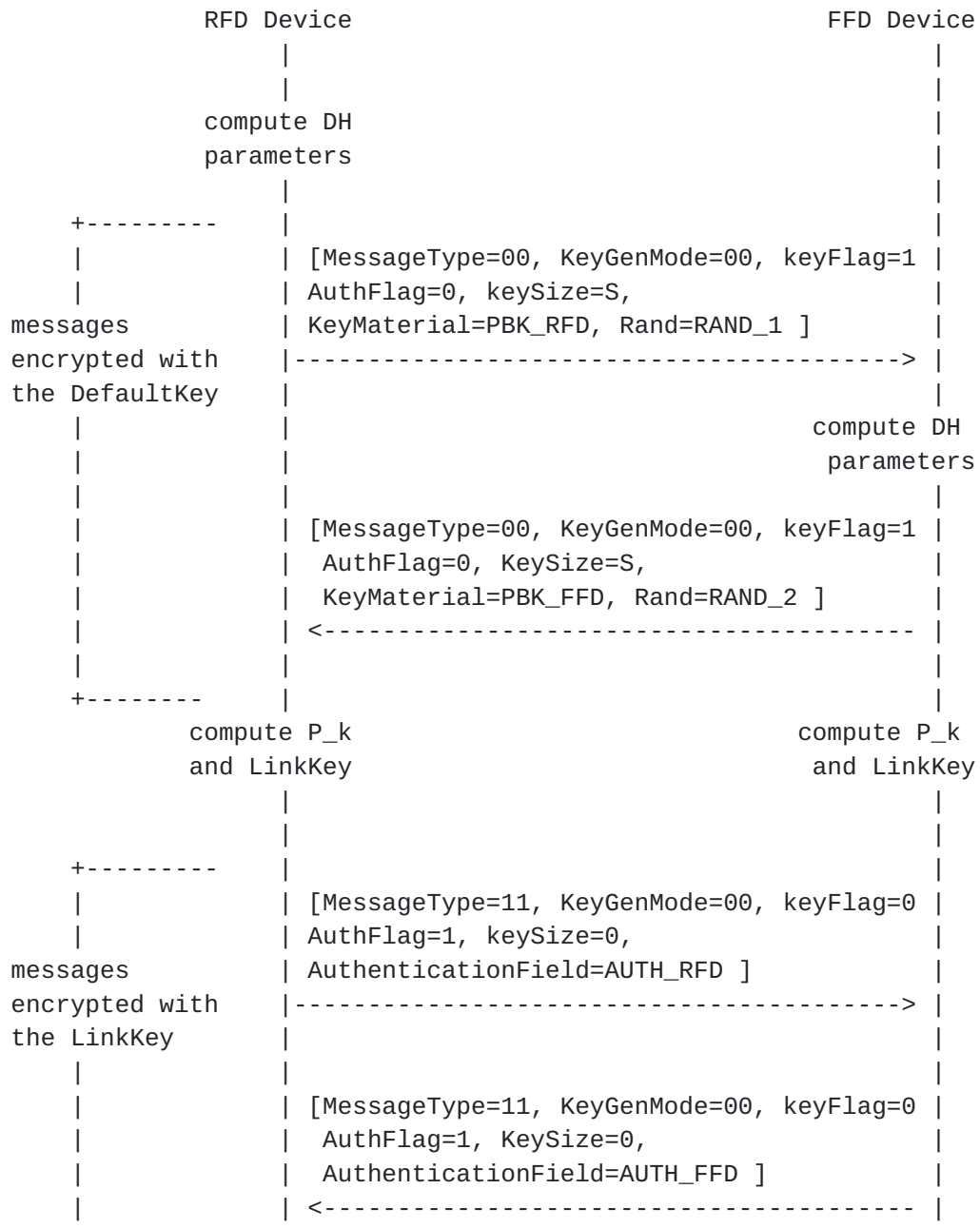

Figure 7. Key exchange mechanism based on DH.

6.5.2 Generation of the LinkKeys

The standard imposes to use the CCM* algorithm and a 128-bit key to protect MAC frames. Independently from the size the PreLinkKey, both RFD and FFD must create a set of link keys, each one 128 bits long. Firstly, each node computes a NewPreLinkKey, NP_k128, through the 128-bit hash function, H_128(.):

NP_k128 = H_128(PAN_ID | P_k).

The NP_k128 key will be used to compute LinkKeys. The CCM* algorithm assumes that each key must be used for a specific number of block ciphers. For each i-th group of block ciphers, the LinkKey, L_k, is computed as in the following:

L_k = H_128 (i | PAN_ID | P_k).

Finally, both FFD and RFD devices update their MAC security attributes by using the procedures described in Secs. 6.5.3 and 6.5.4, respectively.

6.5.3 Update of MAC security attribute for the FFD node after the generation of the LinkKey

After the calculation of the i-th LinkKey, the FFD updates its MAC security attributes as described in what follows.

a) If i=1, a new DeviceDescriptor element, associated to the RFD node with which it has negotiated a link key, is created and stored into the macDeviceTable. It is composed by:

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 RFD 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 RFD node.

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

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

b) The FFD creates the keyDescriptor associated to the i-th LinkKey, L_k, by following these steps:

b.1) A new KeyIdLookupList data structure is created and stored within the KeyDescriptor element. 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 RFD node that initialized the key negotiation phase, 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]).

b.2) 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.

b.3) The DeviceDescriptorHandleList is created and populated with the pointer to the DeviceDescriptor created before.
b.4) The i-th portion of the LinkKey, L_k, is stored within the Key field.

c) The KeyDescriptor created in the previous step is added to the macKeyTable.

6.5.4 Update of MAC security attribute for the RFD node after the generation of the LinkKey

After the calculation of the i-th LinkKey, the RFD updates its MAC security attributes as described in what follows:

a) A keyDescriptor associated to the i-th LinkKey, L_k, is created by following these steps:

a.1) a new KeyIdLookupList data structure is created and stored within the KeyDescriptor element. 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 RFD node, 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]).

a.2) 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.

a.3) The DeviceDescriptorHandleList is created and populated with the pointer to the DeviceDescriptor associate dto the coordinator.

a.4) The i-th portion of the LinkKey, L_k, is stored within the Key field.

b) The KeyDescriptor created in the previous step is added to the macKeyTable.

7 Additional features

There is the possibility to switch from the Flexible Secured to the Hybrid Secure configuration.

To this aim, during the association phase, a device without security capabilities sends to the coordinator a Beacon Request message with the SecurityEnabled flag set to FALSE.

The FFD device then switches to the Hybrid Secure configuration and update all the MAC security attributes accordingly.

From this moment on, the coordinator sends control messages in clear.

8 Security Considerations

There are no security considerations for this document.

9 IANA Considerations

There is no IANA action required for this document.

10 References

10.1 Normative References

[KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC1776] Crocker, S., "The Address is the Message", RFC 1776, April 1 1995.

[TRUTHS] Callon, R., "The Twelve Networking Truths", RFC 1925, April 1 1996.

10.2 Informative References

[EVILBIT] Bellovin, S., "The Security Flag in the IPv4 Header", RFC 3514, April 1 2003.

[RFC5513] Farrel, A., "IANA Considerations for Three Letter Acronyms", RFC 5513, April 1 2009.

[RFC5514] Vyncke, E., "IPv6 over Social Networks", RFC 5514, April 1 2009.

Authors' Addresses

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

Email: g.piro@poliba.it

G. Boggia
DEI, Dep. of Electrical and Information Engineering
Politecnico di Bari
Via Orabona 4, 70125, Bari, ITALY
Phone: +39 0805963913

Email: g.boggia@poliba.it

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

Email: a.grieco@poliba.it