GROUP KEY MANAGEMENT IN IEEE 802.15.4 WIRELESS NETWORKS

by

Gamze Yurttutan

BSc, in Electrical & Electronics Engineering, Middle East Technical University, 2001

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering
Boğaziçi University
2006

ACKNOWLEDGEMENTS

First, I would like to thank my supervisor, Prof. M. Ufuk Çağlayan, for his continuous support and for his many insightful comments during the writing of my thesis. This thesis would not have been possible without his guidance and knowledge.

I would also like to thank my parents, Vicdan & Ibrahim Yurttutan who always encouraged me to complete an M.S. degree. They always believed in me and supported my goals, and provided everything I need to achieve my objectives.

Finally, I would like to thank my friends Özhan Öztürk, Tuğba Demirci, Ceren Atmaca and Sare Sarı for their support, which became my determination.

ABSTRACT

GROUP KEY MANAGEMENT IN IEEE 802.15.4 WIRELESS NETWORKS

This thesis concentrates on establishing a secure group key management scheme in low data rate wireless personal area networks, namely IEEE 802.15.4 standard. In applications where the transmitted data in group communication is sensitive, security of the data should be provided using cryptographic techniques. Security and performance aspects of group key management algorithms are analyzed while initially distributing the group key to all members and redistributing the keys when a member joins or leaves the network.

Security of a group key management means that the algorithm should not let the non-group members to have or guess the group key, while performance of the algorithm depends on many factors. First performance attribute is the speed of the algorithm for fast distribution of keys in an efficient way. Then, regarding that the wireless components' computations cause it to consume power, and this is an unwanted affect, less computational need is important. And finally, suitability to the standard's needs is another key performance factor.

In order to determine the most suitable key management scheme, which offers the best security and performance to IEEE 802.15.4 networks, previously proposed key management algorithms are analyzed and the results are discussed. This analysis shows that the existing algorithms do not fit IEEE 802.15.4's needs, leading us to propose a new group key management algorithm especially designed for IEEE 802.15.4 networks, namely Hybrid Topology Group Key Management Algorithm (HT-GKMA).

ÖZET

IEEE 802.15.4 DÜŞÜK VERİ HIZLI KABLOSUZ KİŞİSEL ALAN AĞLARINDA GRUP ANAHTARI YÖNETİMİ

Bu tezin konusu IEEE 802.15.4 standardı gibi düşük data oranlı kablosuz kişisel alan ağlarında güvenli bir grup anahtarı yönetimi şeması oluşturmaktır. letilen datanın hassas olduğu uygulamalarda, datanın güvenliği kriptografi teknikleri ile korunmalıdır.

Grup anahtarı bütün üyelere dağıtılır. Yeni üyeler dahil oldukça veya eski üyeler ayrıldıkça anahtar tekrar dağıtılır. Bu esnada güvenlik ve performans özellikleri grup anahtarı yönetim algoritmalarıyla analiz edilir.

Algoritma, üye olmayanların anahtarı elde etmesini veya tahmin etmesini engelleyerek grup anahtarının güvenliğini sağlamalıdır. Algoritmanın performansı ise pek çok faktöre dayanır. İlk performans özelliği anahtarların hızla dağıtımını sağlayan algoritma hızıdır. Kablosuz bileşenlerin işlem yapmalar güç harcamasına neden olur. Daha uzun pil ömrü önemli olduğundan bu istenmeyen bir sonuçtur. Standardın ihtiyaçlarına uygunluk ise başka bir performans faktörüdür.

Performans ve güvenlik özelliklerine göre IEEE 802.15.4 ağları için en iyi anahtar yönetim algoritmasını bulmak üzere daha önce önerilmiş algoritmalar analiz edilip sonuçlar tartışılmıştır. Sonuçta var olan algoritmaların IEEE 802.15.4'ün ihtiyaçlarna uygun olmadığı ortaya çıkmıştır. Bu nedenle özellikle IEEE 802.15.4 standardı için üretilmiş bir grup anahtar yönetimi algoritması olan Hibrid Topoloji Grup Anathar Yönetimi Algoritması (HT-GKMA) önerilmektedir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ACL	Access-control list
AKD	Area Key Distributor
BD	Burmester - Desmedt Protocol
CKA	Conference Key Agreement
ERSP	Expand Searching Ring Protocol
FFD	Full functionality device
FCS	Frame Check Sequence
GDH	Group Diffie-Hellman
GKMP	Group Key Management Protocol
GSC	Group security controller
GSI	Group security intermediary
GTS	Guaranteed time slot
HS	Hydra Server
HT-GKMA	Hybrid Topology Group Key Management Algorithm
LKH	Logical Key Hierarchy
LR-WPAN	Low data rate wireless personal area network
MAC	Medium access control
MFR	MAC Footer
MHR	MAC Header
MIC	Message Integrity Code
PAN	Personal area network
PHY	Physical layer
RFD	Reduced functionality device
WLAN	Wireless local area network
WPAN	Wireless personal area network

1. INTRODUCTION

The group key management problem has been widely studied for many years. A number of architectures have been proposed, comparisons have been made[2] [3]. But the effectiveness of these architectures in real life implemented networks with working standards is seldom studied. aAlso, the studies are made for TCP/IP protocol and other respectively older protocols, but for wireless networks, conformity to group key management is still an issue. This thesis will focus on wireless personal area network standard IEEE 802.15.4.

IEEE 802.15.4 [1] is a wireless personal area network (WPAN) standard. Wireless personal area networks (WPANs) are used to convey information over relatively short distances. Unlike wireless local area networks (WLANs), connections in WPANs involve little or no infrastructure. This feature allows small, power-efficient, inexpensive solutions to be implemented for wide range of devices. IEEE 802.15.4 is a standard for low rate WPAN.

Wireless home security, remote thermostats for air conditioner, remote lighting, drape controller, call button for elderly and disabled, universal remote controller to TV and radio, wireless keyboard, mouse and game pads, wireless smoke, CO detectors, industrial and building automation and control (lighting, etc.) applications can be implemented using IEEE 802.15.4 standard. Moreover, For LR-WPANs, multicasting is an important feature when the real life usage areas are considered. As an example, in a car, a light sensor may send multiple lighting devices that information constantly using multicasting. Or, in a smart house system, components may need to send the information to many devices: a heat sensor to the different components of the heating system and refrigerator, while the refrigerator may send the shopping list to the computer as well as the pager, and so on. For the industrial applications, an electrics company who has the reading meters that use IEEE 802.15.4 and the devices have the capability of multicasting of messages between employees can be given as an example.

Furthermore, IEEE 802.15.4 can also be used in medical-care implementations such as emergency medical care, triage, and intensive care. They can all benefit from continuous vital sign monitoring, especially immediate notification of patient deterioration. Sensor data can be integrated into electronic patient care records and retrieved for later analysis. In a wide range of clinical studies, especially those involving ambulatory or at-home monitoring, wireless sensors would permit data acquisition at higher resolution and for longer durations than existing monitoring solutions. Basic needs for this kind of usage is summarized in CodeBlue project [20]:

1. Multiple receivers: Data from a given patient will typically be received by multiple doctors or nurses caring for the patient. Therefore, the network layer should support multicast semantics.
2. Security: Aside from the obvious security considerations with sensitive patient data, some countries' laws mandate that medical devices meet the privacy requirements of the 1996 Health Insurance Portability and Accountability Act (HIPAA).
3. Device mobility: Both patients and caregivers are mobile, requiring that the communication layer adapt rapidly to changes in link quality. For example, if a multihop routing protocol is in use, it should quickly find new routes when a doctor moves from room to room during rounds.

These kinds of needs separate medical care applications from sensor networks. In a traditional sensor network approach, there exist stationary node deployments that transmit data at relatively low data rates, with a focus on best-effort data collection at a central base station. By contrast, medical monitoring requires, reliable communication, and multiple receivers (e.g. PDAs carried by doctors and nurses). Moreover, unlike many sensor network applications, medical monitoring cannot make use of traditional in-network aggregation since it is not generally meaningful to combine data from multiple patients.

These requirements fit IEE 802.15.4 WPAN standard which is not only a sensor network standard but a network environment for different capability devices where the main object is to transmit periodic small data to other hosts. In addition to that, another requirement for multicasting is also supported in network layer. The requirement for secure multicasting environment will be the analyzed in this thesis.

Since security is important for multicating environment, and a group key management scheme is needed for secure and efficient management, regeneration, and revocation of the keys; selection for the most suitable key management algorithm is important. There are different key management algorithms, all of which have its pros and cons. Therefore, in this thesis, the existing group key management structures will be analyzed for the suitability of their features to the IEEE 802.15.4 standard and then a group key management algorithm that best answers IEEE.802.15.42's needs will be proposed.

1.1. Problem Definition

Key management in group communication wireless networks is widely studied in terms of security, performance and complexity. Comparisons are made [2] [3], simulations are run and are being used in some wireless networks. IEEE 802.15.4 is a relatively new wireless networking standard, whose MAC and PHY layers were defined by IEEE community. What is needed is the analysis of the group key management algorithms with respect to the IEEE 802.15.4 standard.

The aim of this thesis is to analyze the existing and widely studied key management algorithms for group communication in terms of their appropriateness to IEEE 802.15.4 Low Data Rate Wireless Personal Area Network Protocol and to propose an alternative group key management algorithm for IEEE 802.15.4 networks. Group key management is divided into three subgroups in terms of their organizations; namely centralized, decentralized and distributed. In each group of group key management algorithms, a number of algorithms will be evaluated for their suitability to IEEE 802.15.4 from the point of security and performance.

The proposed algorithm is a hybrid algorithm where different capability devices use different group key management algorithms. Therefore, there is no single key in the multicasting group. Rather, multicasting group are logically divided into subgroups which use different keys with different algorithms. A decentralized algorithm is used for Reduced Functionality Devices, whereas the Full Function Devices use distributed algorithm. All the performance and security comparisons are made with the regarding group key management algorithms and the results are discussed.

In order to achieve this, first a brief background on IEEE 802.15.4 will be given. Then the different group key management algorithms will be introduced. After giving important aspects of the IEEE 802.15.4 protocol while analyzing the key management algorithms, selected algorithms will be analyzed in detail. Finally, an improved algorithm, especially designated for IEEE 802.15.4 networks, will be proposed.

2. IEEE 802.15.4 OVERVIEW

IEEE 802.15.4 defines the physical (PHY) and medium access control (MAC) layers specification for a low data rate wireless personal area network (LR-WPAN). Security services for the WPAN network are incorporated in the MAC layer. The difference from Bluetooth is that 802.15.4 is a low-rate WPAN. Here low rate refers to low data transmission (Maximum network speed is 250 kbit/s) and low power consumption. These two properties play an important role in sensor networks. IEEE 802.15.4 may seem similar with another IEEE standard: IEEE 802.15.1 (Bluetooth), which is another wireless personal area network also operating in the 2.4-GHz unlicensed frequency band. However, Bluetooth is more oriented toward user mobility and eliminating short-distance cabling; IEEE 802.15.4 aims more for grand-scale automation and remote control. Therefore, their usage areas and aims are different.

The IEEE 802.15.4 usage scenario can be visualized using the following example: A hospital has a wireless sensor network that sends the heartbeat values of the patients to a central computer which are monitored by a nurse. The main purpose of the network here is not sending files, it is sending small data regularly.

Security is incorporated into the MAC layer. Hoever, MAC layer security is not be the only security for devices using IEEE 802.15.4. Additional security controls can also be present in the higher layers. Zigbee Specification [16] is an example of a higher level protocol using IEEE 802.15.4 that includes additional services. Throughout this thesis, only security mechanisms incorporated into the MAC layer will be considered.

The MAC layer provides security services such as access control, data encryption, and frame integrity. There are many options that can be chosen by an upper level application ranging from the key sizes to encryption technique, which also affects the power consumption of the WPAN devices. However, there are still some security considerations about IEEE 802.15.4[15].

2.1. General Description of 802.15.4 Network

A system conforming to IEEE 802.15.4 consists of several components. The most basic component is the network device. A network device can be a full-function device (FFD) or a reduced-function device (RFD). The FFD can operate in three modes serving as a personal area network (PAN) coordinator, a subnetwork coordinator, or a device. An FFD can talk to RFDs or other FFDs, while an RFD can only talk to an FFD. An RFD is intended for applications that are extremely simple, such as a light switch or a passive infrared sensor; they do not have the need to send large amounts of data and may only associate with a single FFD at a time. Consequently, the RFD can be implemented using minimal resources and memory capacity.

Two or more devices within a POS communicating on the same physical channel constitute a WPAN. However, a network shall include at least one FFD, operating as the PAN coordinator.

There are two different topologies supported by IEEE 802.15.4, namely star and peer-to-peer topologies.

In the star topology, communication is established between devices and a single central controller called the PAN coordinator. A device has an associated application and is either the initiation or the termination point for network communications. A PAN coordinator may also have a specific application, but it can be used to initiate, terminate, or route communication around the network. The PAN coordinator is the primary controller of the PAN. All devices operating on a network of either topology have unique 64 bit extended addresses. This address can be used for direct communication within the PAN, or it can be exchanged for a short address allocated by the PAN coordinator when the device associates. The PAN coordinator may be mains powered, while the devices will most likely be battery powered. Applications that benefit from a star topology include home automation, personal computer (PC) peripherals, toys, games, and personal health care. Peer-to-peer topology also has a PAN coordinator; however, it differs from star topology in that a device can communicate with any other

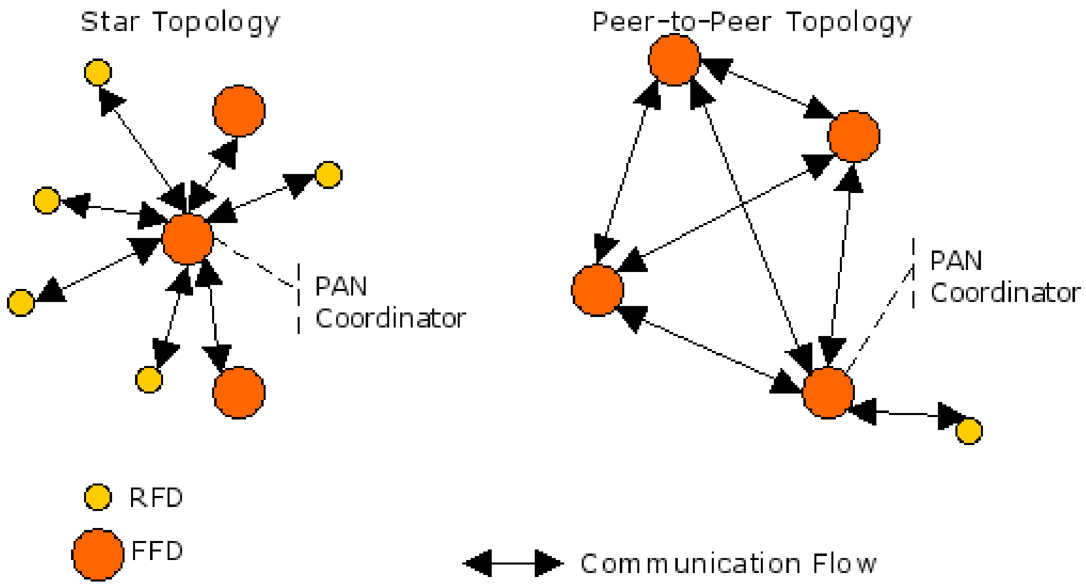


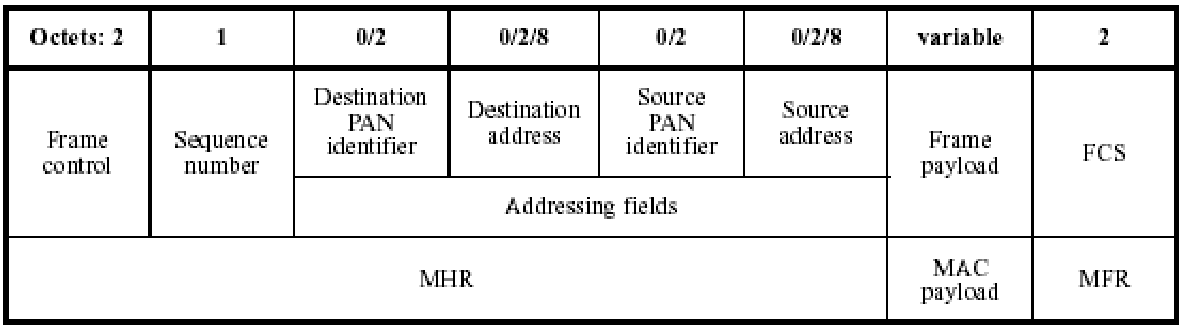
Figure 2.1. Star and Peer to peer topology examples

device as long as they are in each other's range. Peer-to-peer topology allows more complex network formations to be implemented, such as mesh networking topology. Applications such as industrial control and monitoring, wireless sensor networks, asset and inventory tracking, intelligent agriculture as well as security would benefit from such a network topology. A peer-to-peer network can be ad hoc, self-organizing and self-healing. It may also allow multiple hops to route messages from one device to any other device on the network. Such functions can be added at the network layer, which is not defined by the IEEE 802.15.4 standard.

IEEE 802.15.4 can also form hybrid networks which partially contain star and mesh topologies.

IEEE 802.15.4 defines Physical and Medium Access Control Layers of a LR-WPAN network. It should be noted that the network formation is assumed to be performed by the upper layers, therefore is not part of this standard.

Table 2.1. General MAC Frame Format



Octets: 2	1	0/2	0/2/8	0/2	0/2/8	variable	2
Frame control	Sequence number	Destination PAN identifier	Destination address	Source PAN identifier	Source address	Frame payload	FCS
		Addressing fields					
MHR						MAC payload	MFR

2.2. MAC Frame Formats

This section summarizes the different frame formats that IEEE 802.15.4 uses. Each MAC frame consists of the following basic components:

- A MHR which comprises frame control, sequence number, and address information.
- A MAC payload, of variable length, which contains information specific to the frame type. Acknowledgment frames do not contain a payload.
- A MFR, which contains a FCS.

There are 4 different frame types defined in MAC layer. These are Beacon, Data, Acknowledgment and MAC Command Frames.

A General MAC frame is shown in Figure 2.1. The MAC frame format is composed of an MHR, a MAC payload, and an MFR. The fields of the MHR appear in a fixed order, however, the addressing fields may not be included in all frames. A brief explanation of all the fields are given below:

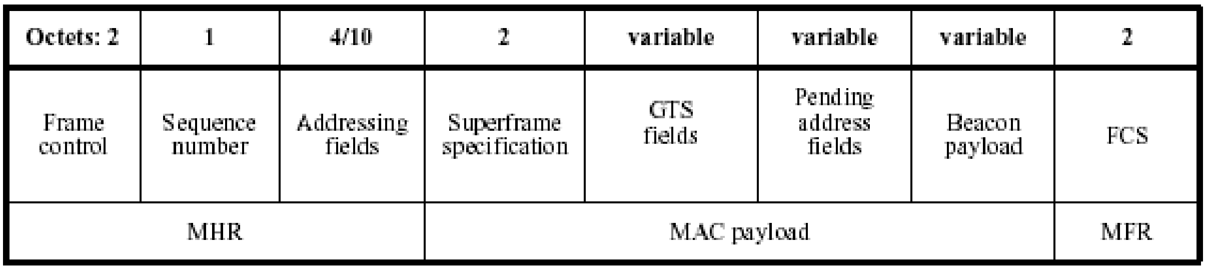
1. Frame Control Field: The frame control field is 16 bits in length and contains information defining the frame type (whether the frame is a Beacon, Ack, Data or MAC command frame), frame security (whether the frame is cryptographically protected by the MAC sublayer), destination and source addressing modes

(whether the address field contains 16 bit short address or 64 bit extended address), PAN destination (whether the MAC frame is to be sent within the same PAN (intra-PAN) or to another PAN (inter-PAN)), ACK request (whether [1] an acknowledgment is required from the recipient device on receipt of a data or MAC command frame)

2. Sequence Number Field: The sequence number field is 8 bits in length and specifies a unique sequence identifier for the frame.
3. Destination PAN Identifier Field: The destination PAN identifier field is 16 bits in length and specifies the unique PAN identifier of the intended recipient of the frame. A value of 0 x ffff in this field represents the broadcast PAN identifier, which is accepted as a valid PAN identifier by all devices currently listening to the channel.
4. Destination Address Field: The destination address field is either 16 bits or 64 bits in length, according to the value specified in the destination addressing mode subfield of the frame control field, and specifies the address of the intended recipient of the frame.
5. Source PAN Identifier: The source PAN identifier field is 16 bits in length and specifies the unique PAN identifier of the originator of the frame. This field is included in the MAC frame only if the source addressing mode and intra-PAN subfields of the frame control field are nonzero and equal to zero, respectively.
6. Source Address Field: The source address field is either 16 bits or 64 bits in length, according to the value specified in the destination addressing mode subfield of the frame control field, and specifies the address of the frame originator.
7. Frame Payload: The frame payload field has a variable length and contains information specific to individual frame types. If the security enabled subfield is set to 1 in the frame control field, the frame payload is protected as defined by the security suite selected for that relationship.
8. FCS (Frame Check Sequence): The FCS field is 16 bits in length and contains a 16 bit ITU-T CRC. The FCS is calculated over the MHR and MAC payload parts of the frame.

[1] If the security enabled subfield is set to 1, the frame shall be protected using the keys stored in the MAC PIB for the security relationship indicated by the current frame.

Table 2.2. Beacon Frame Format



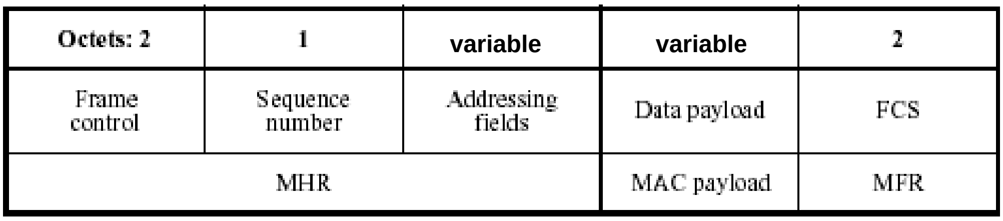
Octets: 2	1	4/10	2	variable	variable	variable	2
Frame control	Sequence number	Addressing fields	Superframe specification	GTS fields	Pending address fields	Beacon payload	FCS
MHR			MAC payload				MFR

All of the different frame formats (i.e. Beacon, data, Ack and MAC Command) comply with the general MAC frame format. However, their MAC payloads are generally different.

Figure 2.2 shows a Beacon Frame format. Here, again there is a frame control, sequence number fields and addressing that are explained before. MAC Payload now contains specific information in order to control superframe structure. The LR-WPAN standard allows the optional use of a superframe structure. The format of the superframe is defined by the coordinator. The superframe is bounded by network beacons, sent by the coordinator, and divided into 16 equally sized slots. The beacon frame is transmitted in the first slot of each superframe. If a coordinator does not wish to use a superframe structure, it may turn off the beacon transmissions. The beacons are used to synchronize the attached devices, to identify the PAN, and to describe the structure of the superframes. Any device wishing to communicate during the Contention Access Period (CAP) between two beacons competes with other devices using a slotted CSMA-CA mechanism. All transactions are completed by the time of the next network beacon. A beacon-enabled network is used for supporting low-latency devices, such as PC peripherals. If the network does not need to support such devices, it can elect not to use the beacon for normal transfers. However, the beacon is still required for network association.

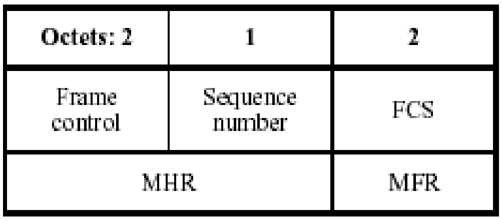
For the Data Frame, MAC payload contains only a data payload, as shown in Figure 2.3. The payload of a data frame contains the sequence of octets that the next higher layer has requested the MAC sublayer to transmit. If security is required on an outgoing data frame, the sequence of octets in the data payload field is processed

Table 2.3. Data Frame Format



Octets: 2	1	variable	variable	2
Frame control	Sequence number	Addressing fields	Data payload	FCS
MHR			MAC payload	MFR

Table 2.4. Acknowledgment Frame Format



Octets: 2	1	2
Frame control	Sequence number	FCS
MHR		MFR

according to the security suite corresponding to either the destination address or *macCoordExtendedAddress* if the destination address field is not present.

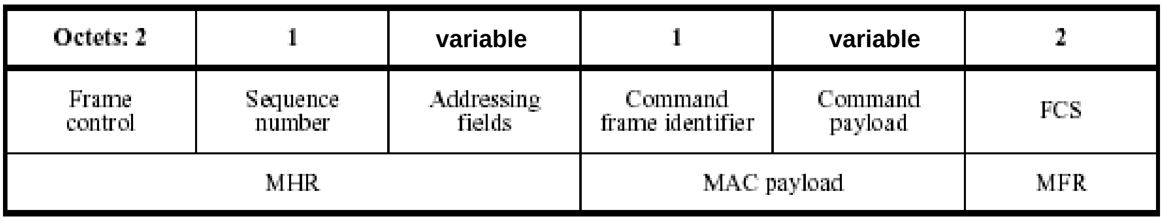
As it is shown in Figure 2.4, acknowledgment frames do not contain a MAC payload. They are used for confirming successful frame reception.

MAC command frames (Figure 2.5) are used for handling all MAC peer entity control transfers. The command frames defined by the MAC sublayer are listed in Figure 2.6.

An FFD is capable of transmitting and receiving all command frame types, while the requirements for an RFD are indicated in Figure 2.6. MAC commands are only transmitted in the CAP for beacon-enabled PANs or at any time for nonbeacon-enabled PANs.

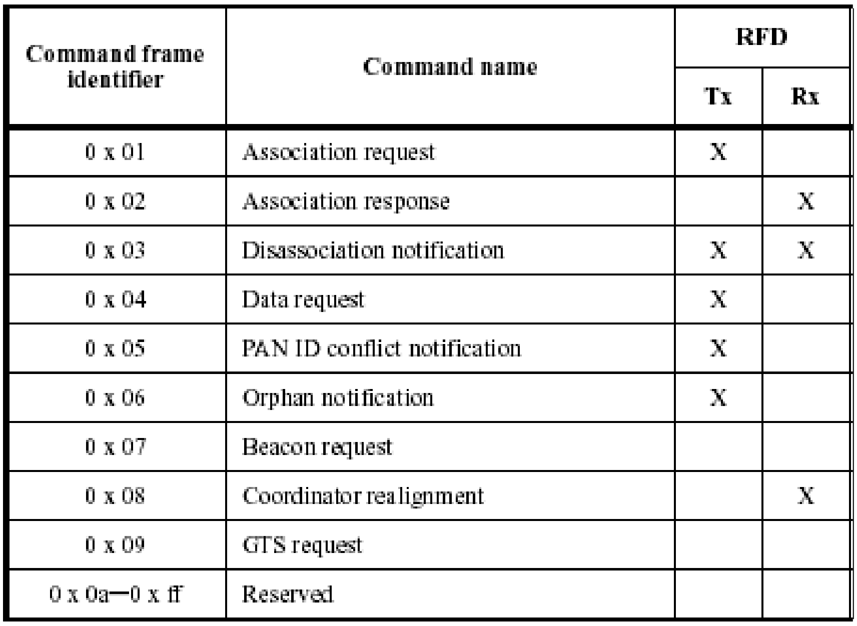
The association and disassociation commands are used to allow devices to associate with or disassociate from a PAN. These commands are association request, association response and disassociation notification commands.

Table 2.5. MAC Command Frame Format



Octets: 2	1	variable	1	variable	2
Frame control	Sequence number	Addressing fields	Command frame identifier	Command payload	FCS
MHR			MAC payload		MFR

Table 2.6. MAC Command Frames



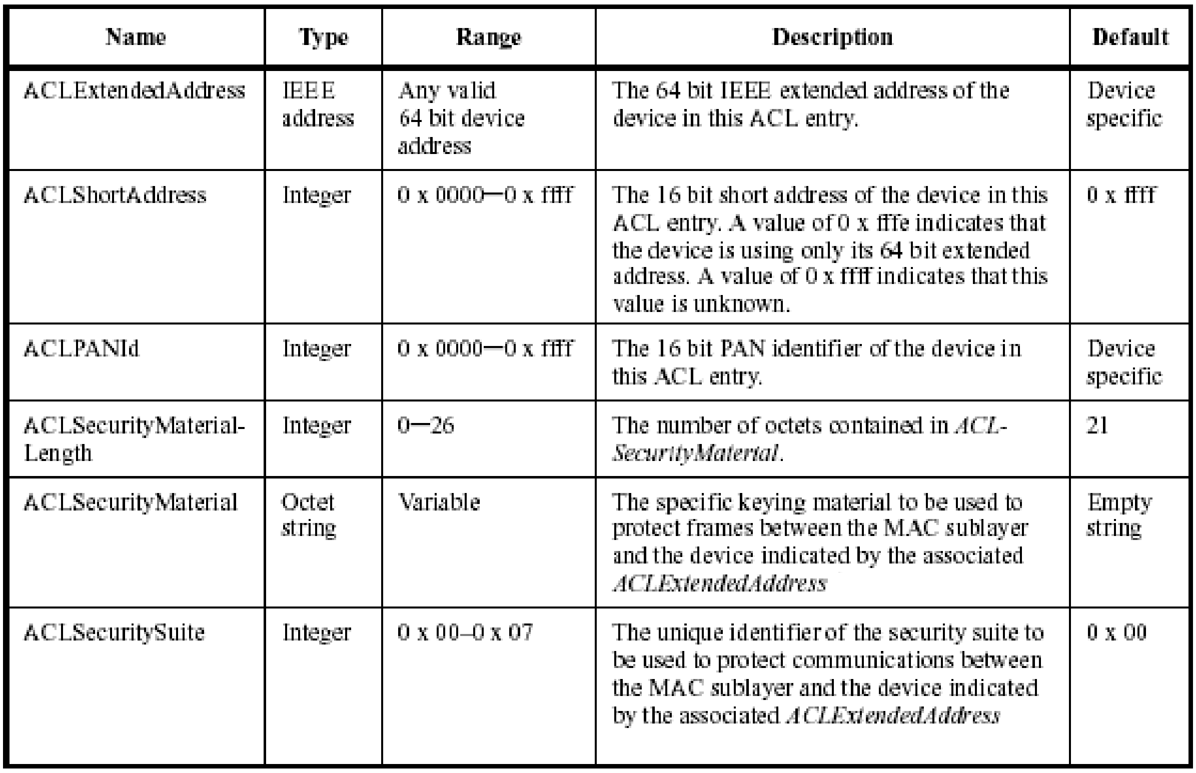
Command frame identifier	Command name	RFD	
		Tx	Rx
0 x 01	Association request	X	
0 x 02	Association response		X
0 x 03	Disassociation notification	X	X
0 x 04	Data request	X	
0 x 05	PAN ID conflict notification	X	
0 x 06	Orphan notification	X	
0 x 07	Beacon request		
0 x 08	Coordinator realignment		X
0 x 09	GTS request		
0 x 0a—0 x ff	Reserved		

The coordinator interaction commands are used to allow devices to interact with a coordinator. These commands are data request, PAN ID conflict association, orphan notification, beacon request and coordinator realignment commands.

The GTS request command is used to manage GTSs. A device can use this command to request the allocation of a new GTS or the deallocation of an existing GTS.

In order to ensure successful operation, MAC layer keeps some constants in its MAC PIB (PAN Information Base). Since the focus of this thesis is group key management in IEEE 802.15.4 networks, the most relevant one is *macACLEntryDescriptorSet*. In this PIB part, the ACL lists are defined, and the total length of the table is written

Table 2.7. ACL entry descriptors



Name	Type	Range	Description	Default
ACLExtendedAddress	IEEE address	Any valid 64 bit device address	The 64 bit IEEE extended address of the device in this ACL entry.	Device specific
ACLShortAddress	Integer	0 x 0000—0 x ffff	The 16 bit short address of the device in this ACL entry. A value of 0 x fffe indicates that the device is using only its 64 bit extended address. A value of 0 x ffff indicates that this value is unknown.	0 x ffff
ACLPANId	Integer	0 x 0000—0 x ffff	The 16 bit PAN identifier of the device in this ACL entry.	Device specific
ACLSecurityMaterial-Length	Integer	0—26	The number of octets contained in *ACL-SecurityMaterial*.	21
ACLSecurityMaterial	Octet string	Variable	The specific keying material to be used to protect frames between the MAC sublayer and the device indicated by the associated *ACLExtendedAddress*	Empty string
ACLSecuritySuite	Integer	0 x 00–0 x 07	The unique identifier of the security suite to be used to protect communications between the MAC sublayer and the device indicated by the associated *ACLExtendedAddress*	0 x 00

in *macACLEntryDescriptorSetSize* attribute. An ACL entry consists of elements that are shown in Figure 2.7.

2.3. Security in IEEE 802.15.4

Although the diverse range of applications to which the standard is targeted imposes significant constraints on requiring a baseline security implementation in the MAC sublayer, some required security functionality is needed in order to provide basic security services and interoperability among all the devices implementing the standard. That includes the ability to maintain an access control list (ACL) and use symmetric cryptography to protect transmitted frames. The higher layers determine when security is to be used at the MAC sublayer and provide all keying material necessary to provide the security services. Key management, device authentication, and freshness protection may be provided by the higher layers.

The security mechanisms in IEEE 802.15.4 are symmetric-key based mechanisms using keys provided by higher layer processes. The management and establishment of these keys is the responsibility of the implementer. The security provided by these mechanisms assume the keys are generated, transmitted, and stored in a secure manner.

2.3.0.1. Access Control. Access control is a security service that provides the ability for a device to select the other devices with which it is willing to communicate. In IEEE 802.15.4, if the access control service is provided, a device maintains a list of devices in its ACL from which it expects to receive frames.

2.3.0.2. Data encryption. Data encryption is a security service that uses a symmetric cipher to protect data from being read by parties without the cryptographic key. Data may be encrypted using a key shared by a group of devices (typically stored as the default key) or using a key shared between two peers (typically stored in an individual ACL entry). Data encryption may be provided on beacon payloads, command payloads, and data payloads.

2.3.0.3. Frame integrity. Frame integrity is a security service that uses a message integrity code (MIC) to protect data from being modified by parties without the cryptographic key. It further provides assurance that data came from a party with the cryptographic key. Integrity may be provided on data frames, beacon frames, and command frames. The key used to provide frame integrity may be shared by a group of devices (typically stored as the default key) or by two peers (typically stored in an individual ACL entry).

2.3.0.4. Sequential freshness. Sequential freshness is a security service that uses an ordered sequence of inputs to reject frames that have been replayed. When a frame is received, the freshness value is compared with the last known freshness value. If the freshness value is newer than the last known value, the check has passed, and the freshness value is updated to the new value. If the freshness value is not newer than the last known freshness value, the check has failed. This service provides evidence that the received data are newer than the last data received by that device, but it does not provide a strict sense of time.

2.3.0.5. Security modes. Depending on the mode in which the device is operating and the security suite selected, the MAC sublayer may provide different security services.

- Unsecured mode: Because security is not used for unsecured mode, no security services are provided by devices operating in unsecured mode.
- ACL mode: Devices operating in ACL mode provide limited security services for communications with other devices. While in ACL mode, the higher layer may choose to reject frames based on whether the MAC sublayer indicates that a frame is purported to originate from a specific device. Because cryptographic protection is not provided in the MAC sublayer in this mode, the higher layer should implement other mechanisms to ensure the identity of the sending device. The service that is provided while in ACL mode is access control.
- Secured mode: Devices operating in secured mode may provide any of the security services defined in the standard. The specific security services are dependent on and specified by the security suite in use. Services that may be provided while in secured mode include access control, data encryption, frame integrity, sequential freshness.

2.4. MAC Layer Security Suite Definitions

There are seven different security modes in 802.15.4 MAC Layer. First one is plaintext mode, in which there is no encryption applied to data. The other modes are AES-CTR, AES-CCM-128, AES-CCM-64, AES-CCM-32, AES-CBC-MAC-128, AES-CBC-MAC-64, AES-CBC-MAC-32.

Security suites may be used when a device is operating in secured mode. A security suite consists of a set of operations to perform on MAC frames that provide security services. The security suite name indicates the symmetric cryptography algorithm, mode, and integrity code bit length. The bit length of the integrity code is less than or equal to the block size of the symmetric algorithm and determines the probability that a random guess of the integrity code would be correct. This bit length does not correspond to the strength of the underlying algorithm. For all security suites in this standard, the algorithm used is advanced encryption standard (AES). Each device that implements security supports the AES-CCM-64 security suite and zero or more additional security suites. Each security suite is specified by a 1 octet value as shown in Figure 3.1. An identifier of 0x00 indicates that secured mode is not to be used.

The counter mode (CTR) symmetric encryption algorithm used in IEEE 802.15.4 standard consists of the generation of a key stream using a block cipher in CTR, with a given key and nonce, and performing an exclusive OR (XOR) of the key stream with the plaintext and integrity code. A nonce is a time stamp, a counter, or a special marker intended to prevent unauthorized message replay. The decryption operation consists of the generation of the key stream and the XOR of the key stream with the ciphertext to obtain the plaintext.

The Cipher Block Chaining Message Authentication Code (CBC-MAC) symmetric authentication algorithm used consists of the generation of an integrity code, using a block cipher in CBC mode computed on a message that includes the length of the authenticated data at the beginning of the data themselves. The verification operation consists of the computation of this integrity code and its comparison to the received integrity code.

The CTR encryption plus CBC-MAC (CCM) combined symmetric encryption and authentication mechanism used consists of the generation of an integrity code followed by the encryption of plaintext data and the integrity code. The output consists of the encrypted data and the encrypted integrity code. The symmetric authentication operation used in this security suite consists of the generation of an integrity code using a block cipher in CBC mode computed on a nonce followed by padded authentication data followed by padded plaintext data if present. The verification operation consists of the computation of this integrity code and comparison to the received integrity code. The symmetric encryption operation used in this security suite consists of the generation of a key stream using a block cipher in CTR with a given key and nonce and performing an XOR of the key stream with the integrity code and plaintext, if present. The decryption operation consists of the generation of the key stream and the XOR of the key stream with the ciphertext to obtain the plaintext and integrity code.

AES is used for encryption for all of the modes. However, Sastry and Wagner [15] pointed out the fact that AES-CTR security mode is so dangerous that it should not even be considered in the IEEE 802.15.4 standard. Use of encryption without a MAC poses significant risk of security breaches. Besides, AES-CTR mode is also susceptible to denial-of-service attacks.

Therefore, it will be assumed that all of our devices (RFD & FFD) use AES-CCM-64 security mode. It is not only default security suite for a device, but also there is no found security problems as in AES-CTR mode. Therefore,the AES-CCM-64 mode of keys will be the focus in analyzing the group key management problem.

2.5. Group Key Management in IEEE 802.15.4

Group key management is an important concept for a wireless network. For example, in a big hospital with different departments such as cardiology, oncology, orthopedics etc. all the departments will have patients and their health information. Assume that there is an IEEE 802.15.4 wireless network in the hospital that sends all the blood pressure, heartbeat, insulin level etc. data. Since these are private information about the patients, they should be protected accordingly. Since network sniffing in wireless networks is easier, a node that is in this wireless network would be able to access the patients' private information. Therefore, it would be appropriate to set up a network in which every department has a different encryption key for the data that they send so that the other departments cannot access their data. That kind of network is possible with the group key concept.

Group keying means grouping the network into communication subsets in which the same key is used. Since the other groups use different keys, each node will be able to decipher only its own group's messages. That will help prevent unintended disclosures, which will eventually increase security.

Creating groups and assigning keys to them, along with the maintenance of the keys throughout their life cycle is stated as Group Key Management. As stated before, IEEE 802.15.4 protocol defines the PHY and MAC layers, and leaves the upper levels. Key management is also a matter for the upper layers, namely for the application layer.

While choosing the right mechanism for the IEEE 802.15.4 network, one should focus on the limitations of the specifications and the maximum benefit for the group key management on 802.15.4. networks. Below are the important considerations while making the choice:

1. FFD and RFD support: IEEE 802.15.4 supports two different types of devices in the network. One is a fully functional device which can perform all the operations while the other is a reduced functionality device which can only communicate with an FFD. This reduced functionality helps the devices to be cheaper with the cost of limited communication environment. We should examine the group management algorithms' handling of RFDs. Obviously; the ideal algorithm should support both FFDs and RFDs.
2. Multiple topology handling: As previously stated, there are different ways of forming the network in a 802.15.4 environment. It could be a star, peer-to-peer or hybrid topology. The behavior of the group management algorithm should be examined under different topologies. It is ideal for the group management algorithm to work with all the topologies.
3. ACL entry optimization: MAC layer has not been designed with group management in mind. Therefore, ACL is designed as if each node will mutually communicate. Each entry holds the key material and the address of the receiver. For instance, if there is group communication between 20 devices, then a single device has to have 19 entries in its ACL. This is a situation to be avoided.
4. Non IEEE 802.15.4 Factors: This section includes group key management pro-

tocol features important for selection but not directly related to IEEE 802.15.4 protocol. These include security features such as forward and backward secrecy, key independence, along with the resistance to known key attacks.

3. GROUP KEY MANAGEMENT ALGORITHMS IN IEEE 802.15.4

3.1. Centralized Group Key Management

In centralized group key management, there is only one entity controlling the whole group. Therefore, in a highly mobile environment, a problem concerning a group controller affects the whole group. Examples include, but are not limited to, Group Key Management Protocol [17], Logical Key Hierarchy, One-way Function Tree, One-way Function Chain Tree, Hierarchical a-ary Tree with Clustering, Centralized Flat Table, Efficient Large-Group Key protocols.

This type of key management protocols are generally suitable for wired networks where members are not mobile ad-hoc. Thus, there is no need for energy-efficient solutions and situations regarding non-availability of the group controller. However, for an IEEE 802.15.4 network where all of the members -RFDs and FFDs- are generally wireless, assigning all of the keying operations to only one member is an important concern. Every single entity should communicate with the group controller. This is a limiting situation for the formation of the wireless network, as in IEEE 802.15.4. Moreover, as the number of group members increases, the management of the group may be a problem. Since the network is wireless, member joins and leaves can be high when compared with a wired network, which makes the management problem even harder for wireless networks. These aspects of central group key management schemes are non-IEEE 802.15.4 factors.

Below is a general inspection of this family of group key management schemes, in terms of their suitability to IEEE 802.15.4:

1. RFD and FFD support: Except for the requirement that the central group controller should be an FFD, there is no limitation for group members. This helps to form WPANs of only one FFD and remaining being RFD, which is a cost-reducing

factor.

2. Multiple Topology Handling: Central Key Distribution, by its nature, requires a star topology for the keys to be transmitted to every member. The situation for a mesh topology network is different. For a mesh topology network which is managed by a central group controller, every single entity should also communicate with the central group controller. There are two options for a device to communicate with PAN coordinator: it either communicates directly, or via another device that supports routing. However, routing is managed by the network layer which is not defined by the IEEE 802.15.4 standard. Therefore, the solution for peer-to-peer networks is dependent on the above protocols and capabilities of the devices. It would not be wrong to assume that a device that has routing capabilities is more expensive. Therefore, it may possibly be costly for a peer-to-peer network to use a centralized group key management mechanism.
3. ACL Entry Optimization: Since the keys are directly distributed via PAN coordinator, ACL entry needs to include the PAN coordinator address for every device, along with every group member.
4. MAC Layer Definitions and Limitations: Since the central group controller is also the PAN coordinator, there does not exist any limitation in terms of MAC layer definitions.

Since the non IEEE 802.15.4 factors have been discussed before, there is no need to mention them again. Because of the aforementioned problems, a centralized group key management protocol is not suitable for an IEEE 802.15.4 wireless personal area network. Therefore, this family of protocols will not be examined in detail. The main focus will be on decentralized and distributed types of key management architectures.

3.2. Decentralized Group Key Management

Decentralized group key management architectures generally split the group into subgroups and key management is done with a different controller in every subgroup. This minimizes the problem of assigning management to a single entity. With this approach, more entities are allowed to fail before the whole group is affected. Exam-

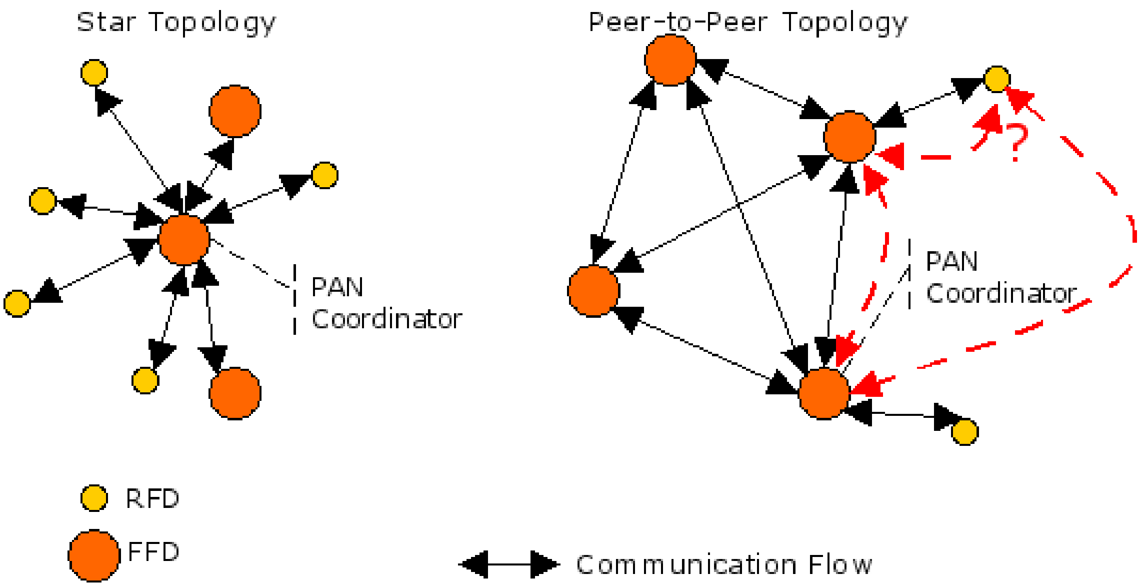


Figure 3.1. Routing in centralized systems

ples for this approach are Scalable Multicast Key Distribution, Iolus, Dual-Encryption Protocol, Cipher Sequence, Kronos, Intra-Domain Group Key Management, Hydra protocols.

In this section, these protocols will be reviewed and their non-IEEE 802.15.4 factors will be commented on. Then suitable protocols will be selected and investigated in detail according to their compatibility to the IEEE 802.15.4 protocol.

Iolus [4] is a protocol that splits the large group into smaller subgroups and assigns a Group Security Agent (GSA) to every subgroup. GSAs are hierarchical and are managed by a Group Security Coordinator (GSC). Iolus is scalable. That is, if a change occurs in one of the subgroups, it does not affect the other subgroups. This protocol will be examined in detail in the following chapters.

3.2.1. Iolus

3.2.1.1. Background. Iolus[4] describes itself as a secure distribution tree multicast. It is composed of a number of smaller secure multicast subgroups arranged in a hierarchy

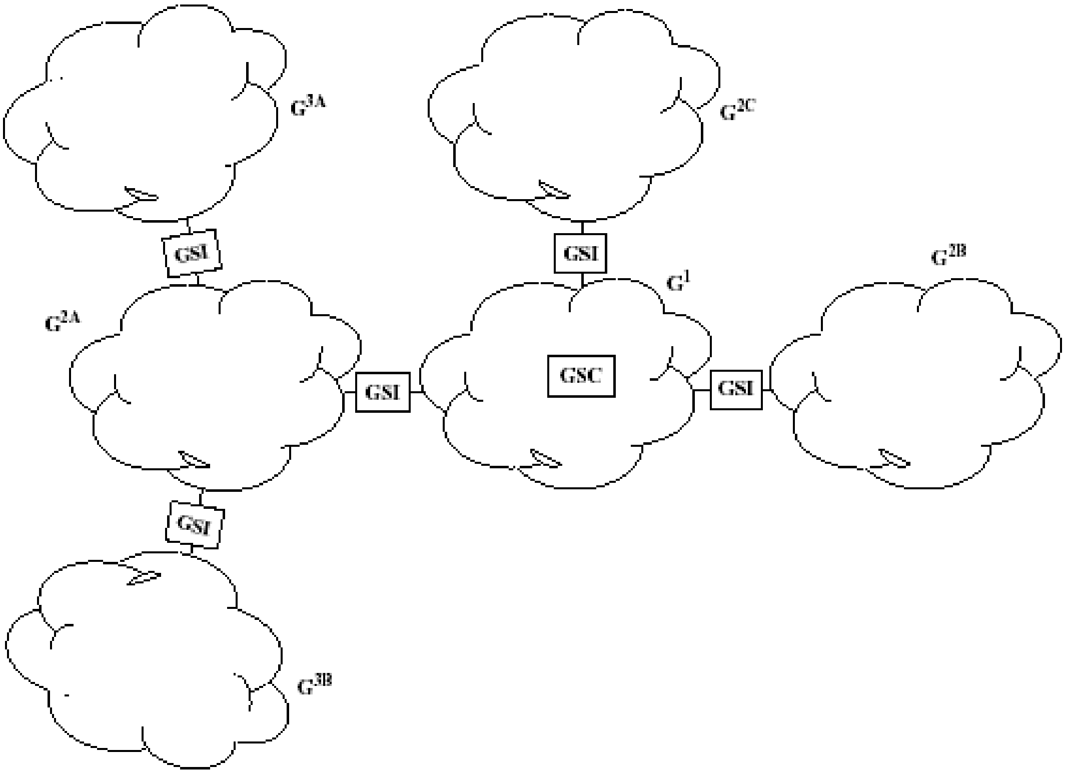


Figure 3.2. Example of a Secure Distribution Tree

to create a single virtual secure multicast group. In this way it aims to enhance scalability issues related with multicasting.

Scalability is achieved by having each subgroup be relatively independent. When a member joins or leaves, it joins or leaves only its subgroup. As a result, only the local key needs to be changed. In order to form the single multicast group from the subgroups, Iolus has Group Security Intermediaries (GSIs) which have the role of connecting subgroups, and Group Security Controllers (GSCs) which manage the top-level subgroups and GSIs (Figure 3.2).

The startup, joining, refreshing, leaving, data transmission, re-keying and shutdown operations of Iolus are detailed in [4].

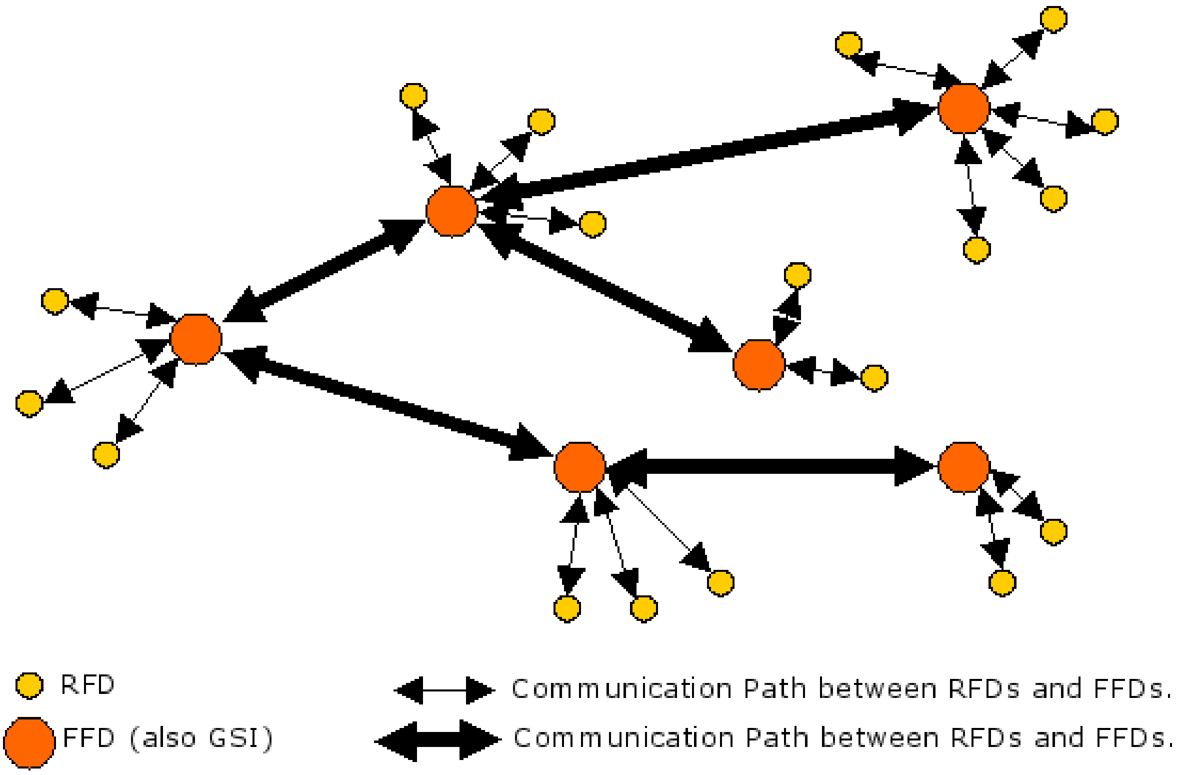

RFD
FFD (also GSI)
Communication Path between RFDs and FFDs.
Communication Path between RFDs and FFDs.

Figure 3.3. IEEE 802.15.4 organization for IOLUS in a tree network topology

3.2.1.2. IEEE 802.15.4 Implementation of Iolus. It is apparent that GSIs and GSCs should be FFDs. Figure 3.3 shows the organization of an IEEE 802.15.4 network using Iolus for group communication. The tree structure of the network can be easily recognized from Figure 3.3.

3.2.1.3. FFD & RFD support in Iolus. As seen in Figure 3.3, Iolus can support both FFDs and RFDs. Not all the devices need to be FFDs in order to set up a group communication scheme. The limitation of Iolus here is that the GSIs and the GSCs should be FFDs. This is reasonable in the sense that RFDs can only communicate with FFDs, so a reasonable amount of FFDs are needed in the network.

Therefore, this feature is supported by Iolus.

3.2.1.4. Multiple Topology Handling in Iolus. Figure 3.3 and 3.4 show different topology layouts of an IEEE 802.15.4 network using both FFDs and RFDs. Figure 8 shows

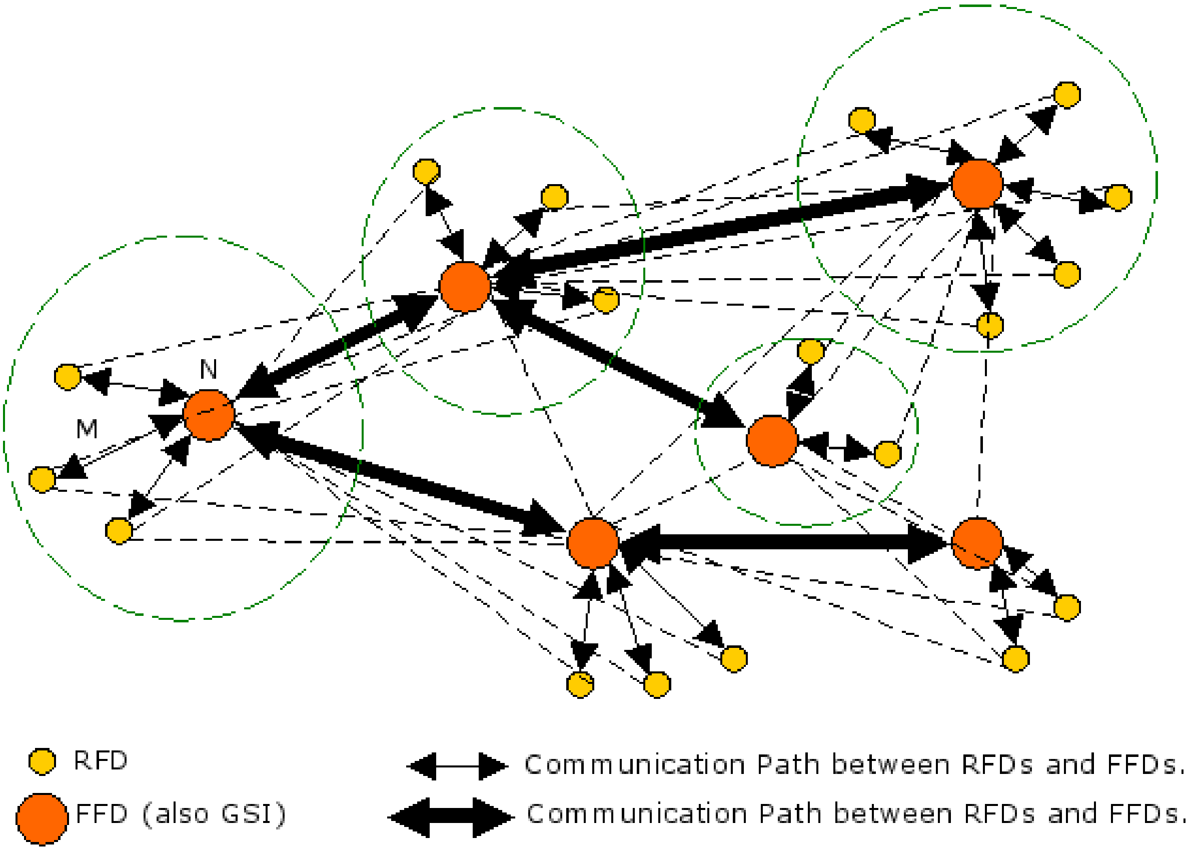

N
M
RFD
FFD (also GSI)
Communication Path between RFDs and FFDs.
Communication Path between RFDs and FFDs.

Figure 3.4. Possible pathways having a mesh topology IEEE 802.15.4 network

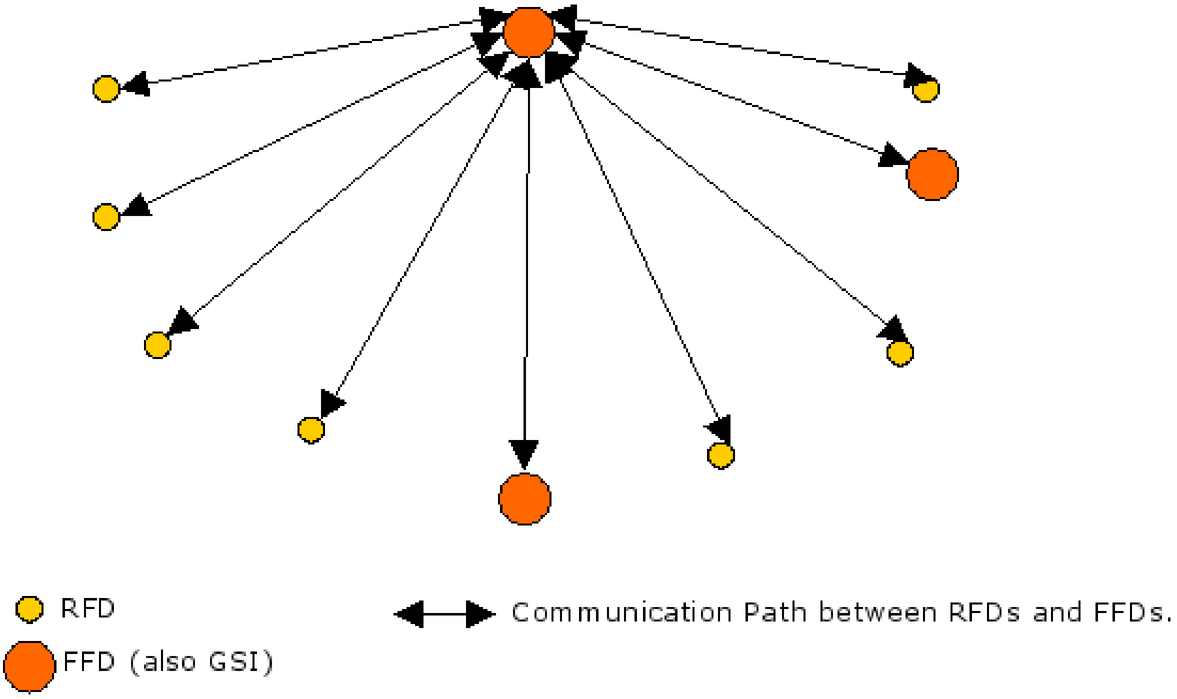


Figure 3.5. IEEE 802.15.4 star topology (No subgroups can be created for Iolus)

a tree network layout in an IEEE 802.15.4, where each RFD is communicates with one FFD, and all the FFDs are hierarchically connected with each other. This is a scheme into which Iolus fits perfectly. The methodology also defines itself as *secure distribution tree* [4].

On the other hand, in a mesh topology, where every FFD is able to send messages, what happens to the group communication & group key management? The answer lies in the ACL of the MAC layer. Even if the network layer is defined to be a mesh topology, if it is not reflected to the MAC, the peers cannot communicate. Iolus is, by definition, tree structured. If all the GSIs have each others' keys, Iolus packets cause bouncing in the network. This can be avoided by ACL rearrangements, according to a minimum spanning tree algorithm, and then Iolus can work. The disadvantage of this method is that the peer-to-peer network topology will be inapparent. The network will behave as if there is a tree topology for key distribution. The virtual tree network will function as the basis for Iolus.

In a star topology network, Iolus becomes meaningless. The latter's aim is to divide the network into subgroups in order to decrease management problems caused by joining and leaving operations; it needs a proper division of subgroups. In order to have subgroups, it needs GSIs. However, a star topology has only one FFD that communicates with every RFD, which is insufficient for creating subgroups. Therefore, even though Iolus can be used in a star topology assuming tree level=0, no GSIs are used and one GSC (which is the FFD) is used, it does not differ from centralized group key management solutions like GKMP [17].

3.2.1.5. ACL list efficiency. In a group communication scheme using Iolus, an RFD should contain the entry for its GSI. Therefore, a single ACL entry containing the subgroup key and the corresponding GSI will be enough. On the other hand, an FFD functioning as a GSI should have its neighboring FFDs and subgroup RFDs in its access control list. In the network described in Table 3.2, FFD "N" should contain 5 ACL entries in order to communicate with all the networks. Table 3.1 and 3.2 shows

Table 3.1. Access Control List of RFD “M”

Access Control List of RFD “M”		
Address	**Key**	**Description**
N	$K_{subgroup}$	GSI key entry
Default	$K_{default}$	Default entry

Table 3.2. Access Control List of FFD “N”

Access Control List of FFD “N”		
Address	**Key**	**Description**
FFD_1	$K_{subgroup1}$	GSI key entry
FFD_2	$K_{subgroup2}$	GSI key entry
RFD_{sg0-1}	$K_{subgroup}$	Subgroup member entry
RFD_{sg0-2}	$K_{subgroup}$	Subgroup member entry
RFD_M	$K_{subgroup}$	Subgroup member entry
Default	$K_{default}$	Default entry

the ACL list of both devices.

In a network with 24 nodes, a complete network covering with 5 ACL entries can be considered to be efficient.

The other members of the network have similar ACLs. Since Iolus is based on a tree structure, each RFD will have only one ACL entry, which is an efficiency increasing advantage.

3.2.2. Hydra

3.2.2.1. Background. Hydra is a scaleable decentralized architecture which creates and distributes symmetric cryptographic keys to large multicast groups [5].Hydra does not employ a manager for subgroup managers. Hence, it is not vulnerable to the failures of single entities.The Hydra architecture is composed of two hierarchical levels. The

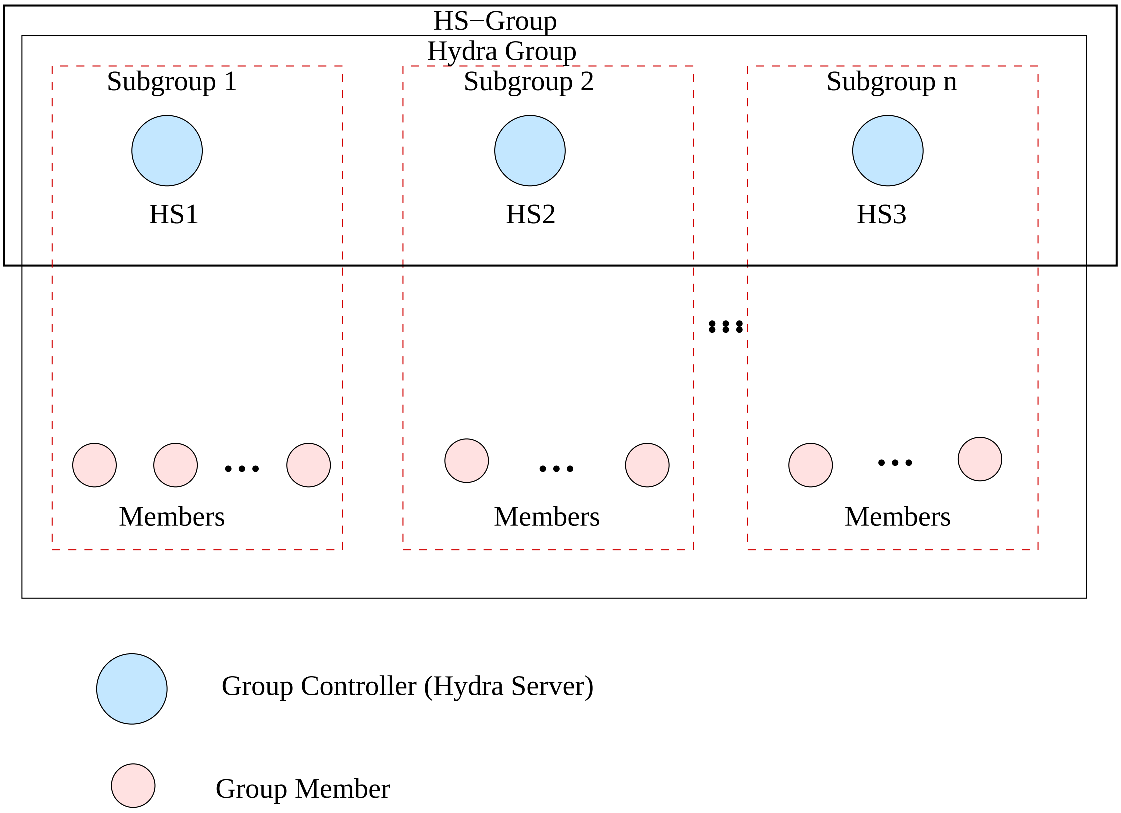


Figure 3.6. Hydra System

top level is composed exclusively of Hydra servers (HSs) (subgroup managers). Group members are placed in the bottom level, separated into subgroups (Figure 3.6).

A group key, namely K_G, protects group communication. All members in the group share K_G. Hydra uses two secure multicast groups for the key management. The first one, called HS-group, is used by the HSs in the top level to agree on a common group key. The second one, called Hydra-group, is used by the HSs to distribute the agreed key to their respective subgroup members in the bottom level. A key called HydraKey protects the communication among the HS-group. HydraKey is shared by all HSs managing a session. A subgroup key protects the GroupKey within a subgroup in the Hydra-group. Each subgroup has its own subgroup key. It is shared between the HS controlling the subgroup and all subgroup members. When a new Group- Key is distributed among the HSs controlling a group, the HSs encrypt the new group key with their respective subgroup key and then send it out.

Hydra makes it possible for all HSs managing a group session to change the group key. When a membership change takes place at an HS and a new key must be generated, it can generate the new group key and send this key to the other HSs involved in that session. The possibility of one or more HSs crashing does not interfere with the remaining HSs.

The membership changes induce rekey operations in order to provide forward and backward confidentiality. Then, the HSs relay the new group key to their respective subgroup members. One or more HSs being unavailable does not interfere with the remaining HSs. Members associated with a failing HS can rejoin the group session by simply executing the ERSP protocol and connecting to another HS.

Hydra employs a Public Key Infrastructure (PKI) model to authenticate all parties in the system. The PKI root certification authority is the group creator. The GC's certificate is the root of the hierarchy. The GC also maintains three other certificates. Each certificate is used to issue a specific set of certificates.

3.2.2.2. IEEE 802.15.4 Implementation of Hydra. In an IEEE 802.15.4 network, all of the HS-Group members should be FFDs. This places a restriction on the formation of the network. Figure 3.7 shows a complex hybrid IEEE 802.15.4 where the multicast group uses Hydra. There are also non-members of the multicast group, which is also drawn. The Hydra Group Controller can be the PAN Coordinator. The Hydra Servers (HS) need to communicate with each other, so they are FFDs. The other network components can be RFD or FFD, as seen in Figure 3.7.

The main disadvantage of using Hydra in IEEE 802.15.4 networks is the need of using a different encryption type, namely PKI inside HS-Group. Since the multicasting group messages are being encrypted in the MAC level using the predefined encryption schemes in the protocol definition, using PKI in the MAC level is impossible. What can be done to push the Hydra protocol in IEEE 802.15.4 networks is to use it in upper layers, namely the application layer. In this way, a Hydra System can be set up. The

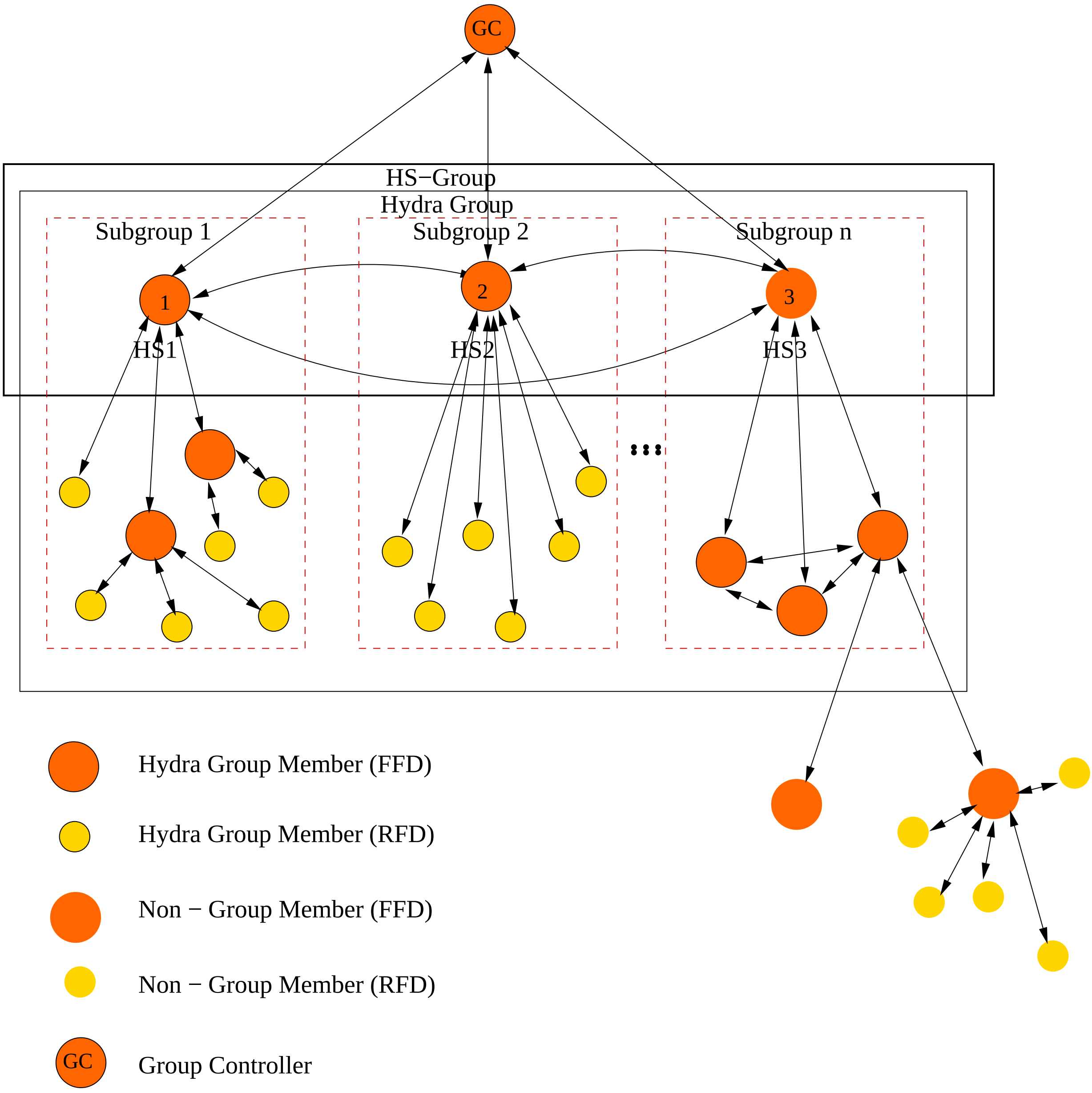


Figure 3.7. Hydra System in an IEEE 802.15.4 Network

drawback of this approach is that the HS-Group communication will not be encrypted from the MAC layer but from upper layers. However, group communication will still be encrypted from the MAC layer.

3.2.2.3. FFD & RFD support in Hydra. RFDs are supported in hydra, as seen in Figure 3.7, but there is a limitation: the HS group devices and GC should be FFD. Other than this, the multicasting group members can be RFD, and they can be directly connected to HSs.

3.2.2.4. Multiple Topology Handling in Hydra. When the network is setup as a star topology network, separating members (except for the PAN coordinator) into different multicasting groups using Hydra is meaningless. Since there could be only one HS-Group member, which is the PAN Coordinator, all the members should be getting their keys from that HS. Therefore, most of the communication protocols used by Hydra (SGKDP, ERSP) are obsolete. This makes Hydra obsolete for star topology.

For the mesh and hybrid networks, Hydra can be used, noting that some of the FFDs should be in HS-Group, meaning that they will have additional duties in group key management.

3.2.2.5. ACL list efficiency. In terms of ACL list efficiency, each Hydra member should hold the group key for every other member, which makes 20 key in a multicasting group with 20 members, including the default key.

For the HS-Group, there should also be asymmetric PKI keys for each member since Hydra uses PKI in HS-Group. But, for the MAC layer of the IEEE 802.15.4 protocol, PKI is not supported. Therefore, for a PKI encryption scheme, upper layers (application layer) should be used. As a result, the HS-Group communication is not encrypted on the MAC level. The keys, therefore, are not held on the ACL list. However, the keys should still be kept within the device.

3.2.3. Kronos

3.2.3.1. Background. Kronos [7], is based uponon the idea of periodic group re-keying. If a group is re-keyed after each membership change, as the size of the group increases and/or the rate at which members leave and join the group increases, the frequency of re-keying becomes the primary bottleneck for scalable group re-keying. In contrast, Kronos can change its scale to handle large and dynamic groups because the frequency of re-keying is independent of the size and membership dynamics of the group.

Re-keying is needed when a member joins/leaves the group, and the frequency of group re-keying depends upon two factors: (i) the size of the group, and (ii) group membership dynamics, i.e., the rate at which members join and leave the group.

Under Kronos, group re-keys are not driven by member join or leave requests. Instead, at periodic intervals, all the member join and leave requests that have accumulated at an AKD (Area Key Distributor) are processed and the new multicast traffic encryption key is securely transmitted to the existing members of the group. An algorithm such as LKH can be used by each AKD to accomplish this task in a scalable manner. Note that most of the processing required for joins and leaves can be done during the time interval between re-keys. Furthermore, under this approach a new traffic encryption key will be transmitted by an AKD to the members in its area even if there has been no membership change during the previous time period.

Two issues need to be addressed for this approach to work correctly. First, all the AKDs must use the same period for re-keying and must have their clocks synchronized so that they re-key at the same time. Second, the AKDs must share some state information that enables them to generate the same key without any communication. Further, no entity other than the AKDs should be able to generate the group key.

The first issue is addressed by having the AKDs agree in advance on the re-keying period and by using a clock synchronization algorithm such as the Network Time Protocol (NTP) [8]. The second issue is resolved if the AKDs agree on two shared secrets (K and R_0before they distribute group keys to members.

Once the shared secrets are established, every AKD generates the multicast group key, R_1, by applying a secret-key encryption algorithm, E, to R_0 using K as the secret key. Thus, R_1= $E_K(R_0)$. R1 is then securely transmitted to the members of the group in the AKD's area.

This process is repeated at each iteration, i.e., the AKD obtains the next multicast group key by applying the secret key encryption algorithm to the the previous group key. Thus R_{i+1}= $E_K(R_i); i \geq 0$

Since the next key is obtained using the current one, there exist forward and backward key secrecy problem. If one of the AKDs is compromised, all of the previous and future keys could be obtained. This is one of the disadvantages of the protocol.

3.2.3.2. IEEE 802.15.4 Implementation of Kronos. Kronos can be implemented with the help of other underlying algorithms such as IGKMP. The only difference is that this time, there exists a periodic rekeying, other than rekeying every time a member is added/deleted. Therefore, the implementation of Kronos is the same with the underlying algorithm. The only requirement here is that all of the parties should be aware of the time, therefore a protocol such as NTP (Network Time Protocol) is needed.

3.2.3.3. FFD & RFD support in Kronos. FFDs may support the extra requirement for the NTP, but RFDs are limited function devices, and by default the standard does not put any obligation for RFDs to support NTP. Even if it does, it may be an extra power consuming asset, which is a disadvantage.

3.2.3.4. Multiple Topology Handling in Kronos. The underlying algorithm handles this part since Kronos is much of a timing based algorithm.

3.2.3.5. ACL list efficiency. ACL list efficiency is another factor dependent on the underlying algorithm.

3.2.3.6. Non-IEEE 802.15.4 Factors. Since we are dealing with wireless networks, the requirement for NTP may be a power consuming burden. It may impose other communication risks when a part of the network loses synchronization with another part. The late distribution of group keys may cause delays in communication.

3.3. Distributed Key Management Protocols

Distributed key management makes it unnecessary to have a group controller for key distribution. Generally, all group members contribute to the generation of the key. This section analyzes this type of key management solutions by giving specific examples and analyzing their suitability to IEEE 802.15.4 architecture.

3.3.1. Group Diffie-Hellman Key Exchange

3.3.1.1. Background. Group Diffie-Hellman Key Exchange protocol [11],[12],[13] uses contributory key generation approach of Diffie-Hellman for generating group keys. Each group member contributes to the key material.

Steiner, Tsudik & Waidner [11] proposed three different Diffie Hellman group key exchange algorithms which they called GDH.1, GDH.2 & GDH.3. Then, using these algorithms, they proposed a group communication architecture, which is called CLIQUES [12]

In this section, these algorithms will be introduced and then the proposed algorithms will be analyzed for usage in an IEEE 802.15.4 environment.

The first algorithm for key distribution is GDH.1, which has two phases: upflow and downflow. In the upflow stage, all the contributions from group members are re-

Table 3.3. Explanation of the notation used

Symbol	*Explanation*
n	number of participants in the protocol
i, j	indices of group members
α	exponentiation base
N_i	random exponent generated by group member M_i
K_n	group key shared among n members

ceived and in the downflow stage, every member computes the group key. For example, in the upflow stage member M_i sends $\{\alpha^{\Pi(N_k|k\epsilon[1,j])} \mid j\epsilon[1, i]\}$ to M_{i+1} for example, M_4 receives the set $\{\alpha^{N_1}, \alpha^{N_1N_2}, \alpha^{N_1N_2N_3}\}$ and forwards to M_5 $\{\alpha^{N_1}, \alpha^{N_1N_2}, \alpha^{N_1N_2N_3}, \alpha^{N_1N_2N_3N_4}\}$. Then, at last, highest numbered member of the group receives the message and computes the group key $(\alpha^{N_1 \ldots N_{n-1}})^{N_n}$.

Then the downflow stage begins from M_n, going downward. Every M_i makes i exponentiations: one to compute K_n and (i-1) to provide intermediate values to subsequent group members. For example, assuming n=5, M_4 receives downflow message $\{\alpha^{N_5}, \alpha^{N_1N_5}, \alpha^{N_1N_2N_5}, \alpha^{N_1N_2N_3N_5}\}$. First, it uses the last intermediate value in the set to compute K_n and forwards message $\{\alpha^{N_5N_4}, \alpha^{N_1N_5N_4}, \alpha^{N_1N_2N_5N_4}\}$ to M3.

One of the disadvantages of GDH.1 is the relatively large number of rounds. In order to reduce this, GDH.2 has been proposed. In this algorithm, upflow is the same except all M_i also have to compute a cardinal value. Then, in the downflow stage M_n broadcasts the intermediate values to all the group members. This reduces the number of rounds nearly by half. For example, assuming n=5, member M_4 receives the set $\{\alpha^{N_1N_2N_3}, \alpha^{N_1N_2}, \alpha^{N_1N_3}, \alpha^{N_2N_3}\}$ from member $M_3$and sends to M_5 the set $\{\alpha^{N_1N_2N_3N_4}, \alpha^{N_1N_2N_3}, \alpha^{N_1N_2N_4}, \alpha^{N_1N_3N_4}, \alpha^{N_2N_3N_4}\}$. The cardinal value here is $\alpha^{N_1N_2N_3N_4}$. When the message reaches M_n, the cardinal value becomes $\alpha^{N_1 \ldots N_{n-1}}$, which can be used to form the group key. Then, in the downflow stage, M_5 broadcasts the set $\{\alpha^{N_1N_2N_3N_5}, \alpha^{N_1N_2N_4N_5}, \alpha^{N_1N_3N_4N_5}, \alpha^{N_2N_3N_4N_5}\}$ so that all the members can compute the group key.

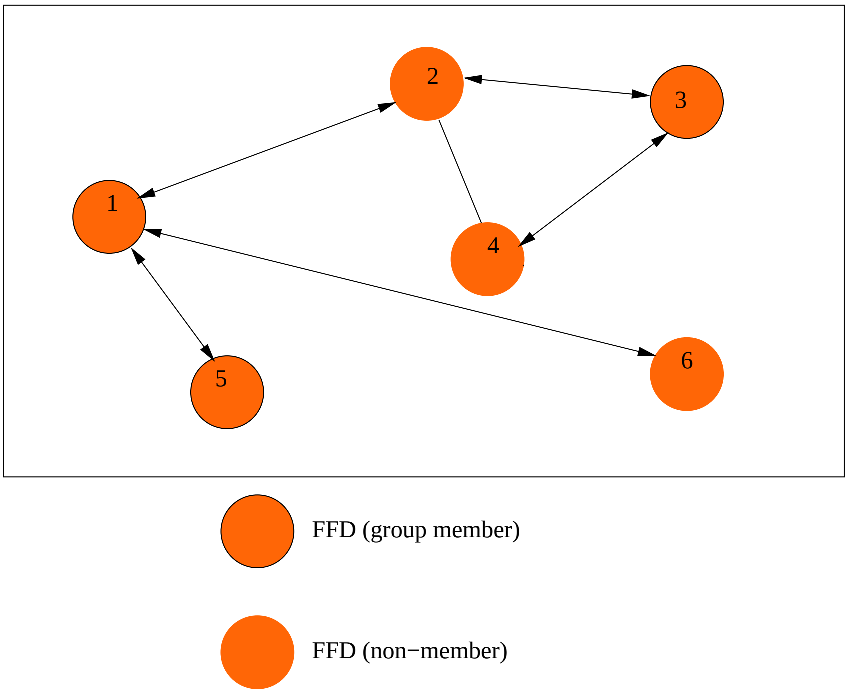


Figure 3.8. IEEE 802.15.4 topology where group uses GDH

GDH.3 is a little different then GDH.1 and GDH.2. In this protocol, there are 4 stages. The first stage is the upflow stage, in which M_i sends $\alpha^{\Pi\{N_k|k\in[1,i]\}}$ where $i \in [1, n-2]$. The second stage is the broadcast stage where M_{n-1} broadcasts $\alpha^{\Pi\{N_k|k\in[1,n-1]\}}$ to all i. In the response stage (third stage) M_i sends $\alpha^{\Pi\{N_k|k\in[1,n-1]\wedge k\neq i\}}$. In the last stage, another broadcast occurs where M_n sends $\{\alpha^{\Pi\{N_k|k\in[1,n]\wedge k\neq i\}} \mid i \in [1, n-1]\}$. In this way, constant message sizes and constant number of exponentiations are achieved. When the number of group members increases, GDH.3 becomes more efficient. In terms of the total number of exponentiations, when $n > 5$, GDH.3 requires less computation.

Member addition and member deletion algorithms have also been defined in GDH.2 and GDH.3 [11].

3.3.1.2. IEEE 802.15.4 Implementation of GDH. Figure 3.8 shows an IEEE network where GDH is used for the distribution of the group communication key. In the topology, nodes circled with black are the members of the multicasting group and the other nodes are not members of the group. Therefore, they must not obtain the group key.

For the sake of simplicity, the network is formed using fully functional devices. The topology for the network is hybrid topology where nodes 1,2,5,6 form a star topology and the nodes 2,3,4 form a mesh topology. It should be noticed that there is no direct path to group member *3*.

The communication flow in this network when the peers are using GDH.2 for group communication, is as follows: Key agreement between M_1, M_3 and M_5 begins with M_1 sending $\{\alpha^{N_1}\}$ to M_3. Since there is no direct route to M_3 from M_1, M_2 receives the message and forwards it to the recipient. After receiving this message, M_3 sends $\{\alpha^{N_1 N_3}, \alpha^{N_1}, \alpha^{N_3}\}$ to M_5, whose possible route is M_2, M_1 and then M_5. Then M_5 computes $\{\alpha^{N_1 N_3}, \alpha^{N_1 N_5}, \alpha^{N_3 N_5}\}$ and broadcasts this message to the whole network. This material is sufficient to enable the group members to obtain the group key, $\alpha^{N_1 N_3 N_5}$.

This is how the GDH is applied on an IEEE 802.15.4 network. It can be understood that when the topology becomes more complex with large number of nodes where there is no mesh topology, it can affect the performance of the algorithm since the rounds are consecutive.

3.3.1.3. FFD & RFD Support. The example shown in Figure 3.8 is a case where all the devices are FFD. But when an IEEE 802.15.4 network is set up with both FFD and RFD devices, the situation becomes different.

Figure 3.9 shows a network where there are 8 group members, 5 of which are RFDs. Nodes inside the red circle are members and the others are the ones which are not supposed to understand the traffic between them.

Now, let us look at the key agreement between them using GDH.2. First of all, M_1 forms its message and sends it to M_2. Then, M_2 forms its message in order to send it to M_3. However, since there is no direct communication path between two RFDs according to the IEEE 802.15.4 standard, it sends its message back to M_1 which

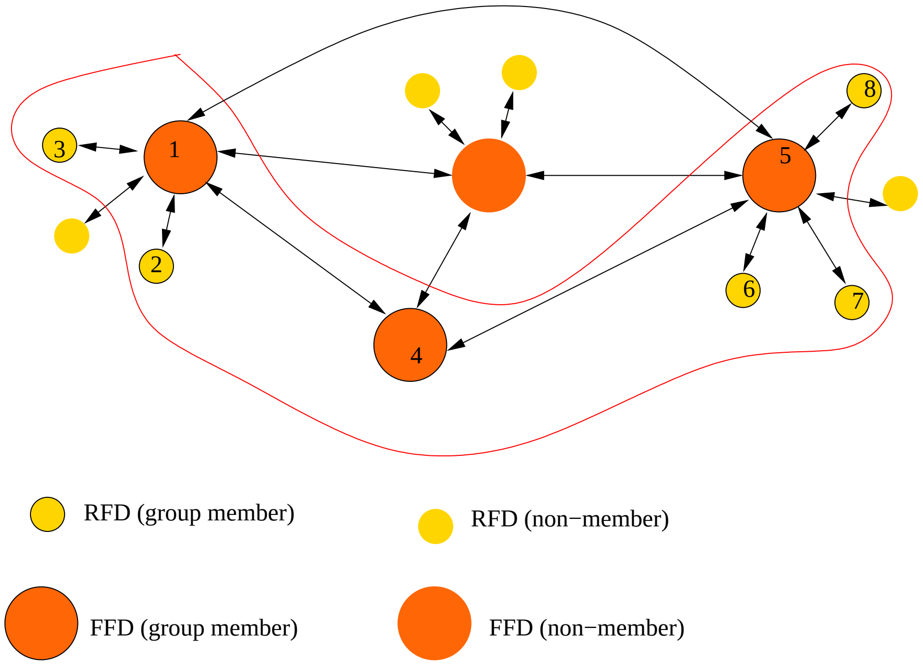


Figure 3.9. IEEE 802.15.4 topology with RFD and FFD

forwards it to M_3. This goes on until M_8 receives the message and forms the broadcast message.

This situation creates a big problem when the number of RFDs is much larger than the FFDs. Even when the RFDs are close to each other, they cannot communicate directly, which increases the work of the FFD. This problem should be avoided if the performance and power consumption of FFD is considered.

3.3.1.4. Multiple Topology Handling. Star topology has much of the same problems discussed in Section 3.3.1.3. In Figure 3.10, a star topology formed with FFD devices is shown on the left. Here, eventhough the devices are capable of communicating with each other, the topology confines them to talk through M_1. This, again, increases the the traffic passing through M_1, causing the device to consume more power. For networks whose nodes are mostly wireless, it is not desirable for one component to discharge rapidly. Therefore, if all the nodes including the center node is wireless, GDH with star topology is not desirable. The same problems also occur in member addition and deletion algorithms after the group is established.

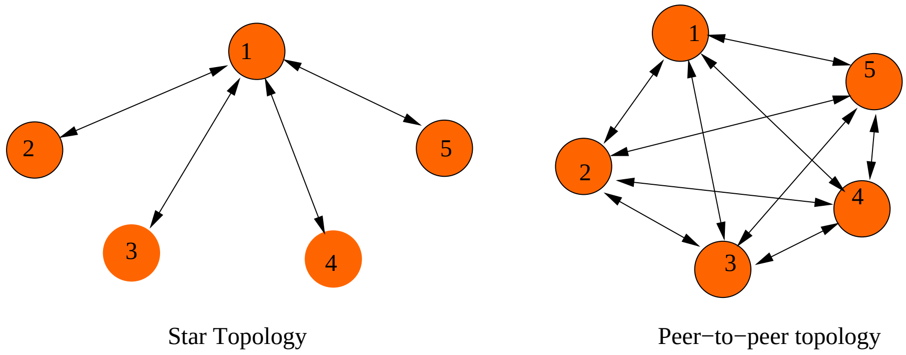


Figure 3.10. IEEE 802.15.4 star and peer to peer topology using GDH for key agreement

For the peer-to-peer IEEE 802.15.4 network in Figure 3.10, there is no problem such as extra forwarding between members since they communicate directly. In fact, for a peer-to-peer IEEE 802.15.4 topology formed only with FFD, all of the comparisons made in [2] apply.

3.3.1.5. ACL list efficiency. Whether the network topology is star or peer-to-peer, the number of ACL list entries does not change for GDH. Every member should keep the same group key for every other member it wishes to communicate with. Therefore, for a multicasting group with 20 members, every node should keep an ACL with 20 entries, including the default entry.

3.3.2. Burmester and Desmedt Protocol

3.3.2.1. Background. Burmester and Desmedt [22] proposed a distributed key management protocol which is executed in three rounds. In the first round, each user M_i generates its random exponent N_i and broadcasts $z_i = \alpha^{N_i}\ mod\ p$. After receiving the broadcast messages, Every M_i computes and broadcasts $X_i = (z_{i+1}/z_{i-1})^{N_i}$. Finally, M_i computes the key $K_n = z_{i-1}^{nN_i}.\ X_i^{n-1}.\ X_{i+1}^{n-2}...X_{i-2}\ mod\ p$ The key defined by this scheme is different from the GDH protocols, namely $K_n = \alpha N_1 N_2 + N_2 N_3 + ... + N_n N_1$. The protocol is proven secure, provided the Diffie-Hellman problem is intractable.

3.3.2.2. IEEE 802.15.4 Implementation of BD. Assuming a hybrid network as in Figure 3.9, the BD protocol can be implemented as follows: Every RFD_i computes its random exponent N_i and sends $z_i = \alpha^{N_i}\ mod\ p$ to its center FFD. Then the FFD broadcasts that message. FFDs directly broadcast their z_i. In the second stage, every FFD and RFD broadcast their X_i in the same way. Then, every member is able to compute the key, K_n.

3.3.2.3. FFD & RFD support in BD. FFDs are fully supported by this protocol. However, for RFDs to take part, RFDs should be capable of computing exponentiation operations. Even if this is possible, having them compute such operations will have negative effects on their battery life. Furthermore, their inability to start a broadcast message directly results in a two-step send operation.

3.3.2.4. Multiple Topology Handling in BD. BD supports mesh topology, as explained above. For the star topology, all the RFD sends their messages to the center FFD, and then the latter broadcasts those messages back to the RFDs. This may overload the FFD, causing an inefficiency.

3.3.2.5. ACL list efficiency. As for the previous protocols, each member should keep the key of every other member since there is no data transformation. This is another problem of the Burmester-Desmedt protocol.

3.3.3. Octopus

3.3.3.1. Background. Octopus [23] protocol is also based on the Diffie-Hellman key exchange protocol. In Octopus, the large group (composed of n members) is split into four sub-groups (n/4 members each). Each subgroup internally agrees on an intermediate DH value: $I_{subgroup} = \alpha^{u_1 u_2 \ldots u_{n/4}}$, where u_i is the contribution from user i and the subgroups exchange their intermediary values. All group members can then calculate the group key. The leader in each subgroup is responsible for collecting contributions from its sub-group members and calculating the intermediary DH value

$(I_s ubgroup)$. Let us call the subgroup leaders A, B, C and D. First, A and B, using DH, exchange their intermediary values I_a and I_b, creating $\alpha^{I_a \cdot I_b}$. Also, C and D do the same and create $\alpha^{I_c \cdot I_d}$. Then, A and C exchange $\alpha^{I_a \cdot I_b}$ and $\alpha^{I_c \cdot I_d}$. Leaders, B and D do the same. Now, all of them can calculate $\alpha^{I_a \cdot I_b \cdot I_c \cdot I_d}$. After that, A, B, C and D send $\alpha^{\frac{I_a . I_b . I_c . I_d}{u_i}}$ where $i = 1...(n-4)/4$, to their respective subgroups and all members of the group are capable of calculating the group key.

This protocol can be extended to a 2^d cube which is the 2^d-octopus protocol. In the 2^d-octopus protocol, the participants act as in the Octopus protocol, the only difference being, instead of four parties 2^d are distinguished to take charge of the central control whereas the remaining $n - 2^d$ parties divide into 2^d groups.

3.3.3.2. IEEE 802.15.4 Implementation of Octopus. For an IEEE 802.15.4 network completely implemented with FFDs in mesh topology, using Octopus is straightforward. Four FFDs are selected as the leaders, and then the others are shared between these four nodes. Then, the groups first form their subgroup intermediary DH value, and then they are interchanged between other leaders. After everyone receives the material, they are sent to the subgroups on a member basis.

However, if the topology is a star topology as in Figure 3.5, then assigning four leaders is not possible. The cube structure over a star topology is not feasible. When the hybrid structures are considered, for some specific topologies, Octopus may be a very effective fit. For example, for topologies that have power of 2 star subtopologies they can be used, assigning the center FFD of the star subtopology as leader. But it does not work for every possibility of topology layouts. For instance, the topology in Figure 3.11 is a completely legal network topology in IEEE 802.15.4, yet it is not supported by Octopus. Here, the fourth leader that is needed by Octopus cannot be found. Therefore the topology limitations in the standard keeps the algorithm from running properly. This is not an acceptable situation for a proper algorithm, so Octopus is not suitable for use within an IEEE 802.15.4 network.

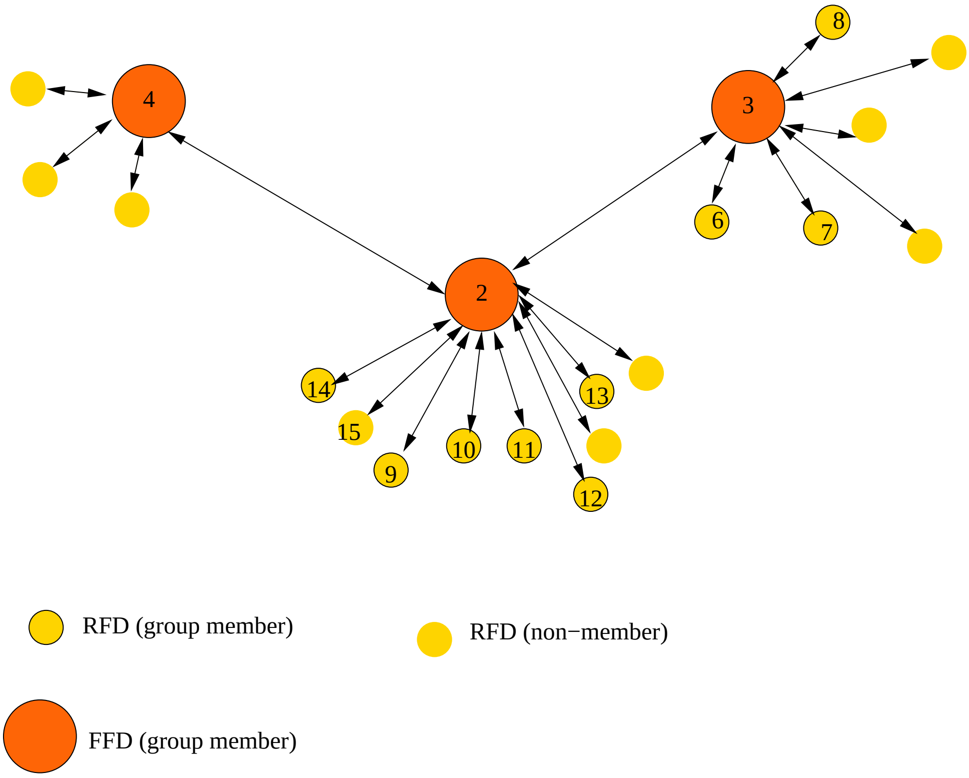


Figure 3.11. Example IEEE 802.15.4 topology

3.3.3.3. FFD & RFD support in Octopus. FFDs are fully supported, while there are some restrictions on RFDs. First of all, RFDs cannot be leaders. Moreover, they should be capable of generating random numbers since their contribution is needed while forming intermediary DH values.

3.3.3.4. Multiple Topology Handling in Octopus. As stated in 3.3.3.2, different possibilities of network formation in IEEE 802.15.4 standard are not fully supported in Octopus although the latter may be efficient in some specific topology examples. Still, a general fitting to the standard is not supported by Octopus. Therefore it can be concluded that it is infeasible.

3.3.3.5. ACL list efficiency. In Octopus, each member should keep the same group key for every other member it wishes to communicate with. This means that for a multicasting group with 20 members, every node should keep an ACL with 20 entries, including the default entry.

3.3.4. CKA

3.3.4.1. Background. Boyd [24] proposed another protocol for conference key agreement (CKA) where all group members contribute to generating the group key. The group key is generated with a combining function: $K = f(N_1, h(N_2), ..., h(N_n))$, where f is the combining function, h is a one-way function, n is the group size, and N_i is the contribution from group member i. The protocol specifies that $n1$ members broadcast their contributions (N_i) in the clear. The group leader, for example U_1, encrypts its contribution (N_1) with the public key of each member and broadcasts it. All group members who had their public key used to encrypt N_1 can decrypt it and generate the group key.

3.3.4.2. IEEE 802.15.4 Implementation of CKA. When CKA is used in IEEE 802.15.4 networks, members first generate their contribution N_i and broadcast it, except for the assigned group leader which can be the PAN coordinator in this case. Then, the PAN coordinator encrypts its contribution. But here, if PKI is used, a MAC level security is not possible since only AES with different modes are supported in the standard at MAC level. In order to fully obey the CKA algorithm, the encryption that PAN coordinator uses should be done in the upper layers. Moreover, the final message that is broadcasted is a large message which is the combination of encrypted value N_1 with every member's public key. Since the IEEE 802.15.4 standard is a low data rate standard, it has a small packet size, which may require the total message to be fragmented. After the fragmentation, more than one packet will need to be transmitted.

It should also be noted that, for an RFD to send its contribution to the other members, it should first send it to its FFD. But that FFD will probably have other members with the same request. Moreover, it should also transmit the broadcast messages coming from other members. Therefore, the FFDs connected to a star subtopology as the center, may become bottlenecks.

Finally, every member, after receiving the required material, should compute the group key. The function is a combining function of all the contributions from every member. Therefore, when the group size increases, the power consumption costs may be high, especially for RFDs to compute the key.

3.3.4.3. FFD & RFD support in CKA. FFDs and RFDs are both supported by the algorithm, but it should be noted that the algorithm puts computational burden on RFDs, which they may not be able to handle. As stated before, in order to take part in this algorithm, RFDs should send their contribution to the central FFD. After that point, FFD directs the message.

3.3.4.4. Multiple Topology Handling in CKA. Every kind of topology suits the CKA algorithm. Within a mesh network, the broadcast messages are sent more easily. When it is a star topology, every member sends its contribution to the central FFD which is in this case also the PAN coordinator. Then, the coordinator sends the broadcast messages to every member. Hybrid topologies are also supported since they are simply a combination of the topologies.

3.3.4.5. ACL list efficiency. In CKA, just as in single key group key management schemes, every member should keep the same group key for every other member it wishes to communicate with.

3.4. Evaluation and Comparison of Analyzed Algorithms

If multicasting is a need in an IEEE 802.15.4 network, distribution of the group keys, ensuring group communication security, as well as the efficiency of the member addition and deletion algorithms are the problematic areas. In the previous chapter, different viewpoints on key management in group communication are overviewed. Every algorithm has its pros and cons when used in an IEEE environment and the main bottleneck is IEEE 802.15.4's ability to form the network using various topologies with various kinds of devices (FFDs and RFDs)

When a network is formed using only FFDs and the topology is mesh, the network is in fact equal to an ad-hoc wireless network in which every member is capable of routing messages to other nodes. In this kind of topology, the distributed type of algorithms ensure flexibility. Therefore, it is a good choice to go for them. Among the distributed group key management algorithms, the ones in which all the group members contribute to group key generation are more fault-tolerant and diminish the risks of vicious key generation by a single entity. According to Anton and Duarte [18], the CLIQUES protocol suite gives better results in terms of messages vs. the number of nodes and exponential operations vs. the number of nodes. Therefore, in this thesis CLIQUES will be taken as a basis for the aforementioned type of IEEE 802.15.4 networks.

However, when the topology is star and the devices are RFDs, distributed approaches are not efficient because of the limitations of the star topology. For a star topology network, key management can be effectively done using centralized and decentralized approaches. Since there exists a central management point for communication, that point can also handle the key management issues arising from group communication. But, when a centralized approach is used in the case of member addition and deletion, all the star topology members should change the keys. Therefore, a local re-keying approach would be useful especially for large star topology IEEE 802.15.4 networks.

Hence, it can be concluded a hybrid solution is much more suitable for an IEEE 802.15.4 where the mesh connected FFDs use a distributed algorithm while the star connected RFDs' group key management is done by a decentralized algorithm that uses local re-keying. In this way, every possibility of forming an IEEE 802.15.4 is covered, keeping in mind that hybrid networks with both star and mesh topologies can also be set up.

Forming a hybrid approach and covering every possibility adds an overhead to FFDs. They have to keep two different algorithms in mind. Assuming that the FFDs are more capable devices, this approach lowers the computation complexity on RFDs, which is a desirable result. Shifting all the computation and transformations to FFDs helps RFDs be energy efficient.

The proposed group key management scheme that is especially designed for IEEE 802.15.4 will be discussed in detail in the following chapter.

4. A NEW GROUP KEY MANAGEMENT SCHEME FOR IEEE 802.15.4 NETWORKS

Group communication is generally a way for transmitting multimedia such as videos and photos. But establishing a secure group communication scheme in a low data rate wireless personal area network is also important. That this kind of network is intended for transmitting small amounts of data, does not imply that they do not need multicasting. It is, in fact, a need for an IEEE.802.15.4 network as suggested in Code Blue project [20] where a patient tracking system is established. In this case, periodically controlled patient information which is obtained by sensors such as heartbeat, insulin level etc. , is sent to the attending doctors. What is transmitted is important information that needs to be secured. Therefore, secure multicasting is a need for IEEE 802.15.4 networks.

If a system needs group communication, it should decide how to manage the group keys. Group key management means first to establish an initial key distribution to all members, and then to redistribute the keys when a member joins or a leaves the network. Two important aspects of these procedures are security and performance.

Security of a group key management means that the algorithm should not let the non-group members to have or guess the group key while performance of the algorithm depends on many factors. The first performance attribute is the speed of the algorithm for the distribution of keys. Then, considering that the computations by wireless components result in higher power consumption, less computational need is important. And finally, suitability to the standard's needs is another key performance factor.

In order to decide for the best key management for IEEE 802.15.4 networks according to their security and performance, previously proposed key management algorithms have been analyzed. The results are given in the previous chapter. The conclusion of this analysis is that the existing algorithms do not fit IEEE 802.15.4's needs. For some algorithms, security is questionable while the others do not yield maximum performance since they were not specifically designed for IEEE 802.15.4 networks. Therefore a new group key management algorithm that is especially designed for IEEE 802.15.4 networks, namely Hybrid Topology Group Key Management Algorithm (HT-GKMA) is proposed.

The algorithm covers three situations:

- Initial Key Agreement where each group member receives the group key for first time.
- Member Addition where a new member joins the group and has to receive the group key.
- Member Deletion where a node is no longer a group member and should not have the group key anymore.

Following sections describe the algorithm and the last section compares the proposed algorithm with the existing ones for IEEE 802.15.4 networks.

4.1. Assumptions

The following assumptions are made:

- The IEEE 802.15.4 network topology is hybrid, i.e. mixed mesh and star topology employing both RFDs and FFDs (as depicted in Figure 4.1).
- The PAN Coordinator is aware of the type of the network topology (Hybrid, just star, etc.)
- The PAN Coordinator holds the list of the multicasting group members.
- Every FFD knows the group members. (The PAN Coordinator distributes this list)
- Every star subtopology member has a unicast communication channel with its central FFD. That is, they have a shared key other than the group key that is used for one-to-one communication.

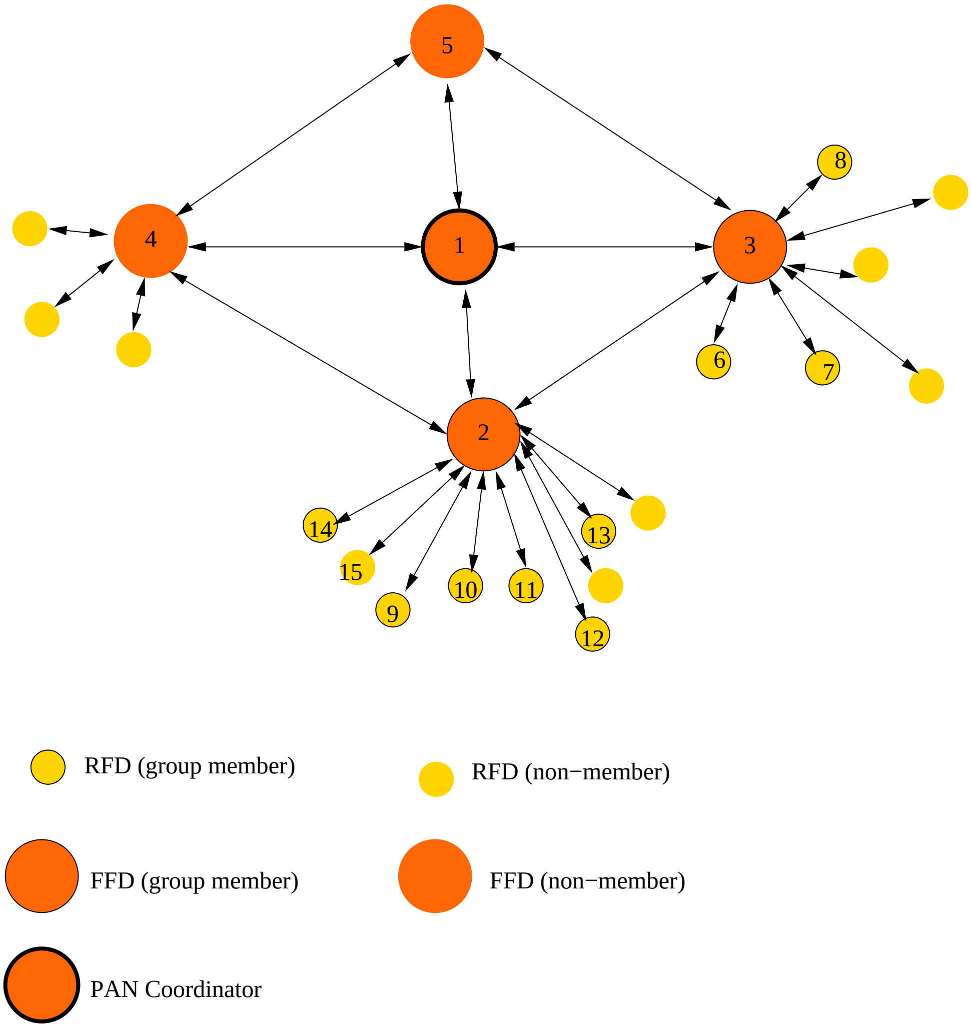


Figure 4.1. IEEE 802.15.4 Network Topology Example

Table 4.1. Notation used in describing proposed algorithm

Symbol	Explanation
FFD_i	i^{th} FFD in the group
α	Exponentiation base
N_i	Random exponent generated by group member FFD_i
n	Total number of FFDs in the group
r	Total number of RFDs in the group
s	Total number of star subtopologies in the group
K_n	Subgroup key for the FFDs
K_{j-f}	Subgroup key for the subgroups in the star subtopology
j	Indices for the different star subtopologies in the network
l_j	Total number of members in the j^{th} star subtopology in the existing IEEE 802.15.4 network
m	Maximum subgroup member number (constant)
FFD_{str_j}	Center FFD in the j^{th} star topology
j_f	Indices for the different subgroups in the j^{th} star subtopology
SG_{j_f}	f^{th} subgroup in the j^{th} star subtopology

The notation in Table 4.1 will be used throughout the section.

4.2. Initial Key Agreement

Initial Key Agreement is the distribution of the group key to the group members. Figure 4.2 gives the flow diagram for the initial key agreement. For the initial key agreement to start, the network topology should be known. If the network topology is a pure star topology, then there are no members which need GDH.3 to be used. Therefore, the procedure for distributing keys to a star topology starts directly. The keys are produced and distributed to the subgroups in the star topology as stated below.

In the proposed algorithm, there is not just one key for each member. Instead,

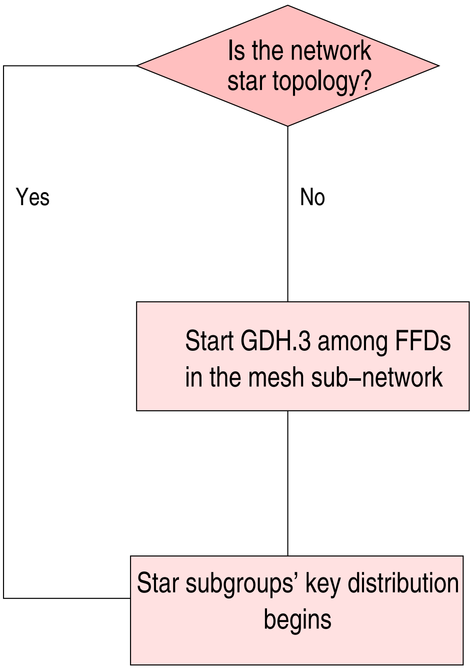

Is the network star topology?
Yes
No
Start GDH.3 among FFDs in the mesh sub−network
Star subgroups' key distribution begins

Figure 4.2. Initial Key Agreement Flow Diagram

different capability and topology devices are separated from each other with subgroup keys. As an example, FFDs forming a mesh network share a subgroup key that is different from the members of the group who are forming star subtopology which are generally RFDs (FFDs in the star subtopology also behave as RFDs. Therefore, while referring the star subtopology members, it should be noted that they may also be FFDs although the notation excludes them for the purpose of simplification). Regarding these factors, the procedure is as follows:

In the first phase, the group member FFDs in the mesh subnetwork form their group key using GDH.3:

1. FFD_i sends $\alpha^{\Pi\{N_k|k\in[1,i]\}}$ where $i \in [1, n-2]$.

2. FFD_{n-1} broadcasts $\alpha^{\Pi\{N_k|k\in[1,n-1]\}}$ to all i

3. FFD_i sends $\alpha^{\Pi\{N_k|k\in[1,n-1]\wedge k\neq i\}}$

4. FFD_n sends $\{\alpha^{\Pi\{N_k|k\in[1,n]\wedge k\neq i\}} \mid i \in [1, n-1]\}$.

In the second phase, star subtopologies have to form their key. But first, each star subtopology should form its subgroups according to the following:

1. Assuming that the total number of members in a star topology is l_j for the $j^{th} star subtopology$, $\lceil l_j/m \rceil$ subgroups will be formed.
2. Every formed subgroup will have m members.
3. The last subgroup does not need to have exactly m members, the remaining members form the last subgroup.

As an example, if l=23 and m=7, there will be 4 subgroups, 3 of which will have 7 members and 1 of which will have 2 members.

After forming the subgroups, the subgroup keys are distributed as follows:

1. FFD_{str_j} generates a different subgroup key for each SG_{j_f}. m is an initially assigned value. Normally it is static and does not change. Deciding on the optimum value for m will be discussed later.

2. FFD_{str_j} distributes the key individually to every member l_j using a secure unicast channel. As presented previously, a secure unicast channel is basically an encrypted one-to-one communication method for secure transmission of group keys.

3. Every member stores this key for group communication. For other communication purposes, different key entries should be used.

As an example, for the network in Figure 4.1, FFD_1 initiates the key agreement by sending α^{N_1}. Then, FFD_2 broadcasts $\alpha^{N_1 N_2}$. After this, FFD_1 and FFD_2 send α^{N_2}

and α^{N_1} to FFD_3, respectively. Then, FFD_3 broadcasts $\{\alpha^{N_2 N_3}, \alpha^{N_1 N_3}\}$. After that, every one can compute the group key If it is assumed that m=4, then for FFD_2, two subgroups need to be formed. First, it generates a key, K_{2-1}, for members 9,10,11,12 and distributes this key to them; then, it generates K_{2-2} for members 13 and 14 and sends the key over a secure channel. Concurrently, FFD_3 generates the subgroup key K_{3-1} and sends it to members 6,7 and 8. This completes the key distribution process.

4.3. Member Join

New member joins can be examined using two scenarios:

1. FFD joining the network
2. RFD joining a star subtopology

For the first scenario the member addition procedure is as follows:

1. We assume that FFD_n saves the content of the Broadcast and Response messages (Stages 2 and 3)
2. FFD_n generates a new exponent and with it, computes a new set of sub-keys $\{\alpha^{\Pi\{N_k|k\in[1,i]\wedge k\neq\, j\}} \mid j \in [1, n]\}, \alpha^{N_1*\ldots*N_{n-1}*\widehat{N}_n}$, which is forwarded to the new member FFD_{n+1}.
3. FFD_{n+1} computes the new key $K_{n+1} = \alpha^{N_1*\ldots*N_{n-1}*\widehat{N}_n*N_{n+1}}$ and adds its own exponent to each of the n sub-keys that are received.
4. FFD_{n+1} broadcasts the sub-keys, and members compute K_{n+1}
5. The subgroup keys of the star subtopology remain unchanged.

For the second scenario, a new member addition is carried out as follows:

1. New member joins the first subgroup where $population < m$.
2. The subgroup key of the corresponding subgroup is regenerated and the new key is transmitted to the new member over a secure unicast channel. For the existing members, the new is encrypted with the previous subgroup key and broadcasted.

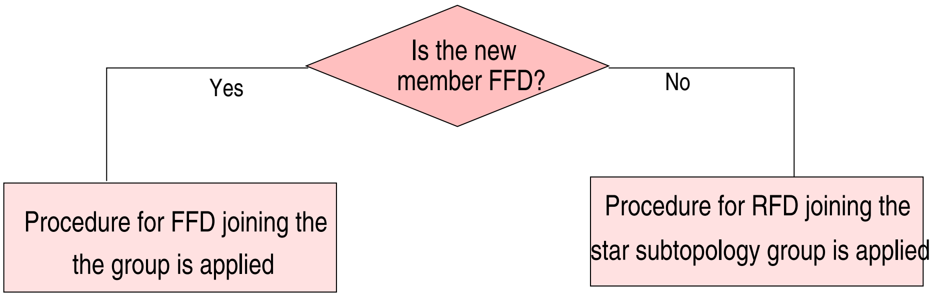

Is the new member FFD?
Yes
No
Procedure for FFD joining the the group is applied
Procedure for RFD joining the star subtopology group is applied

Figure 4.3. Member Join Flow Diagram

3. If all the groups have m members, a new group is formed including only the new member.
4. FFD_{str_j} generates a new key for the new group and sends it to the new member.
5. Other subgroup keys remain unchanged.

Let us examine the procedure on the sample topology. A new member RFD_{15} joins the multicasting group. FFD_2 analyzes the subgroup allocation and places this new member in the first non-full subgroup which, in this case, is the second group. FFD_2 generates K'_{2-2} and sends it to RFD_{13}, RFD_{14}, and RFD_{15}. The other members of the group are not affected from the member addition process and their keys need not change.

On the other hand, if FFD_4 were the one to join the group, the procedure would be much different. In this second case, the procedure is started by FFD_3 which keeps the broadcast message $\alpha^{N_1N_2}$ and the response messages: α^{N_2} and α^{N_1}, which were sent in the initial key agreement. FFD_3 first computes a new exponent N'_3 and then sends $\{\alpha^{N_1N'_3}, \alpha^{N_2N'_3}, \alpha^{N_1N_2}, \alpha^{N_1N_2N'_3}\}$ to the new member FFD_4. Then, using the incoming message, FFD_4 computes $\{\alpha^{N_1N'_3N_4}, \alpha^{N_2N'_3N_4}, \alpha^{N_1N_2N_4}, \alpha^{N_1N_2N'_3N_4}\}$. The last one is the new key. Therefore FFD_4 keeps this value for itself and broadcasts the message $\{\alpha^{N_1N'_3N_4}, \alpha^{N_2N'_3N_4}, \alpha^{N_1N_2N_4}\}$. After receiving this message, FFD_1, FFD_2, FFD_3 are able to compute the new key.

4.4. Member Leave

Member leave will also be analyzed using two scenarios (Figure 4.4. Member deletion for a device leaving the star topology is as follows:

1. FFD_{str_j} deletes the ACL entry for leaving member.
2. FFD_{str_j} generates the new subgroup key for the corresponding subgroup, $\widehat{K}_{j-f}$.
3. FFD_{str_j} sends $\widehat{K}_{j-f}$ to each member over the secure unicast channel.
4. Subgroup members change the group key as $\widehat{K}_{j-f}$.

The procedure for an FFD leaving the network is as follows:

1. If FFD has also been a center for a star topology, every device connected to this FFD should first leave the group according to the previous procedure.
2. Let FFD_p be the member slated for removal from the group, assuming $p \in [1, n-1]$, i.e., $p \neq n$.
3. FFD_n generates a new exponent $\widehat{N}_n$.
4. FFD_n computes a new set of $n-2$ sub-keys: $\{\alpha^{\Pi\{N_k | k \in [1,i] \wedge k \neq\ j\}} \mid j \in [1, n-1] \wedge k \neq p\}$ and broadcasts them to all the group members.
5. In the event that FFD_n is to be removed from the group, FFD_{n-1} assumes the special role as described above.
6. If devices other than FFD_p, who were on the star topology previously, want to join the group again, they should be connected to another FFD which is in the multicasting group, and request member join from that FFD.

4.5. FFD and RFD Support

In the proposed algorithm there is a built-in support for devices. If the group member is a fully functional device (FFD) in a mesh network, the algorithm uses GDH.3 for key distribution, which is more preferable in a wireless environment. On the other hand, when the device is an RFD, meaning that it is in a star subtopology, it is not very suitable to use GDH.3 because of the limitations of the reduced function devices. This

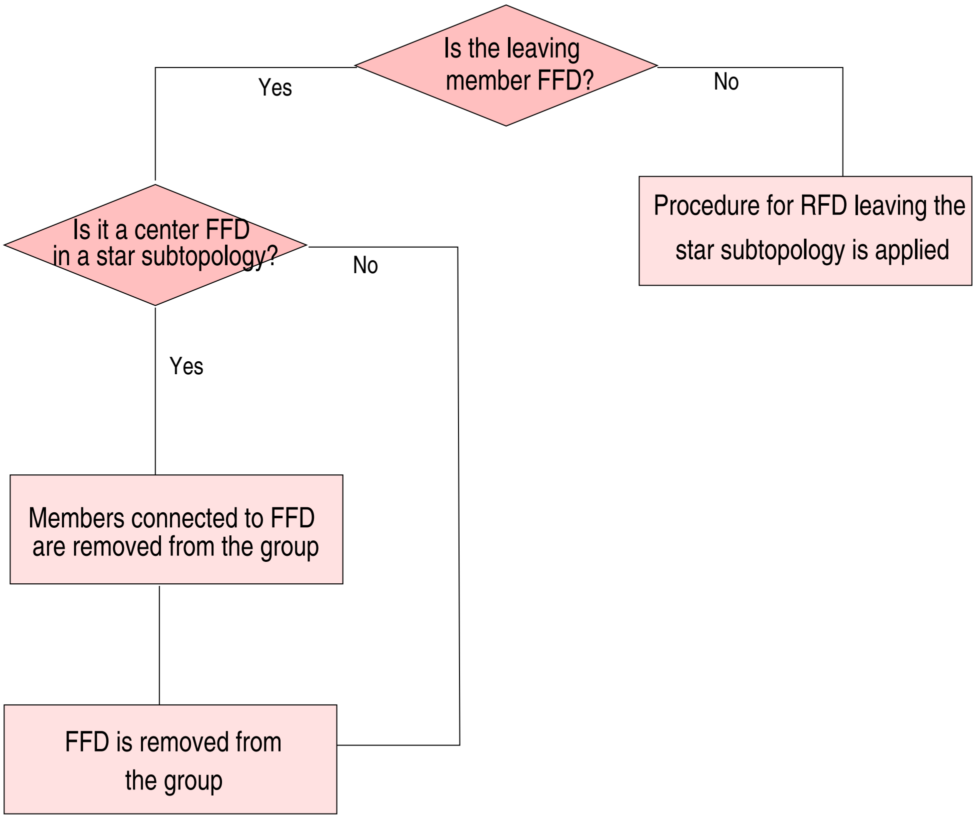

Is the leaving member FFD?
Yes
No
Procedure for RFD leaving the star subtopology is applied
Is it a center FFD in a star subtopology?
No
Yes
Members connected to FFD are removed from the group
FFD is removed from the group

Figure 4.4. Member Leave Flow Diagram

time, the algorithm uses a much simpler methodology for key distribution to RFDs. Their center FFDs in the star topology generate the subgroup key and distribute it to the RFDs and the FFDs in the star subtopology.

Therefore RFDs' limited functionality is effectively used, and they do not need to have the capability of generating random numbers. They also consume less energy, since they do not compute any exponentiations. Hence, the proposed algorithm handles both the RFDs and the FFDs effectively.

4.6. Multiple Topology Handling

The proposed algorithm supports both the star and the peer-to-peer topologies as well as the hybrid topology. When the topology of the network is just a star topology, the algorithm directly jumps to star subtopology key distribution phase, omitting the GDH.3 process for FFDs.

If the topology of the network is fully-mesh, only the GDH.3 phase is applied. If the topology is hybrid, first, the GDH.3 process is applied to the mesh sub-topology. Then, all the star subtopologies concurrently start key distribution.

4.7. ACL Entry Optimization

Since there is a data transformation going from a peer-to-peer to a star topology, there occurs an ACL list optimization, meaning that every group member need not hold an ACL list entry for every other member. There are three possibilities for the ACL lists of the members:

1. A group member who is in the peer-to-peer network and does not conduct a star network holds only the peer-to-peer network member list.
2. A group member who is in the peer-to-peer network and conducts a star network holds the entries for both the peer-to-peer network members and the star network members which are conducted by it.
3. A group member who is a node in the star network only holds the ACL list entry for the FFD conducting that star network.

4.8. MAC Layer Limitations

Everything is dependent on the implementation of the devices but the capabilites of the devices may not be foreseen. However, from a 'best practices' point of view, the main advantage of this hybrid solution is to employ the security features of a Diffie-Hellman approach while not requiring any kind of exponentiations and random number generations on RFDs. This is a more convenient way of implementation in terms of covering most of the products in the market.

The other advantage of this approach is that it imposes minimum computational power on RFDs at the cost of computational burden on FFDs, especially the ones which control a star topology.

4.9. Resistance to Known Network Attacks

This section examines the proposed algorithm's strength on security. When talking about a group key management algorithm's security aspects, there are three key issues:

- Forward key secrecy: Knowing the present key should not lead to the guessing of the future keys.
- Backward key secrecy: Knowing the present key should not lead to the guessing of the previous keys.
- Key independence: None of the generated keys should have any relation with each other.

Forward key secrecy is important when a member leaves the group. It has the last key and since it is no longer a member, this last key should not lead that device to guess the next generated group key after its deletion. For FFDs, the GDH.3 protocol regenerates the keys. This protocol is a natural extension to Diffie-Hellman; it imposes the same security features as the Diffie-Hellman algorithm [12], [11].

Moreover, the other subgroup keys are independently generated when a member leaves the network. The algorithm is randomly generated by the central FFD in the star topology. When a member leaves the group, the subgroup key that is carried by it becomes obsolete since that subgroup key is regenerated by the center FFD of the corresponding star subtopology.

When it comes to backward secrecy, when a new group member joins the group, it should only have the present key, not the past keys. This is a security constraint that a group key management algorithm must support. When a member joins the group, the proposed algorithm first analyzes the capabilities of that device and what kind of subtopology (mesh or star) it is residing in. If it is an FFD that will be connected to the mesh topology, then the member add algorithm of GDH.3 regenerates the subgroup key. Since the new key forming material $\{\alpha^{\Pi\{N_k | k \in [1,i] \wedge k \neq\ j\}} \mid j \in [1,n]\}, \alpha^{N_1 * \ldots * N_{n-1} * \widehat{N}_n}$

is different than $\{\alpha^{\Pi\{N_k | k \in [1,i] \wedge k \neq j\}} \mid j \in [1,n]\}, \alpha^{N_1 * \ldots * N_{n-1} * N_n}$, the new member N_{n+1} cannot guess the previous key.

Furthermore, when a newly added member is about to join one of the star subtopologies, the subgroup key is regenerated and distributed to the existing members using a broadcast message encrypted with the previous group key. The new member receives this key encrypted with its personal key. Hence, the new member does not have access to any material that could compromise the past keys.

Finally, all of the subgroup keys are independent from each other since no one key is used for generating other keys in the network. When a key is needed, it is randomly generated in star subtopologies and every member contributes to key forming in mesh subtopology. The material for key formation in GDH.3 is also randomly generated by each member

4.10. Results

This section discusses the performance and security issues of the proposed algorithm. Before a comparison of the findings with the previously studied algorithms, the optimum value for the constant m that is used in HT-GKMA algorithm will be studied since its value is important for performance. If the chosen value of m is small, then the minimum number of group members will be affected from member join and leave operations. On the contrary, the center FFD for the star subtopology needs to generate and manage a lot of subgroup keys.

However, if the chosen value for m is large, the keys that need to be generated would be less but then, the member joins and leaves to affect more members. An optimum value should be found in order to balance the situation. The chart in Figure 4.5 shows two different charts on top. The first one with the multiple lines represents the number of subgroups that will be formed versus the value of m, with the changing of the total number of members in the star subtopology, l. The values for $l = 20$, $l = 40$, $l = 60$ and $l = 80$ are shown on the chart. As m increases, the number of

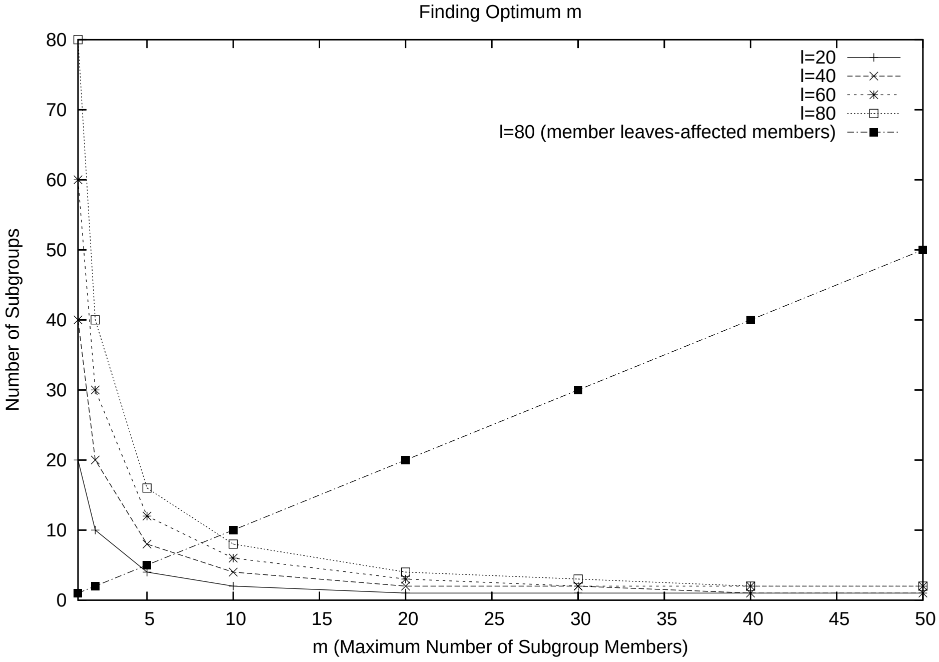


Figure 4.5. Finding optimum m

subgroups decreases. The other chart is to show the affect of m on member join/leave operation. This chart shows the maximum possible number of member number affected by a member leave operation as m increases. It is increasing linearly as m increases, as shown in Figure 4.5. These two charts intersect in between $m = 5$ and $m = 10$ depending on the value of l. Therefore, choosing m between 5 and 10 would be optimum for our algorithm. Throughout the rest of this thesis, it is assumed that m=7 where applicable.

The first comparison aspect is the security of the proposed algorithms. This includes forward and backward key secrecy, as well as key independence. Forward key secrecy means that knowing the present key should not lead to guessing of the future keys. Similarly, backward key secrecy is not being able to guess the previous keys when the present key is known. Finally, the generated keys' not having any relation with each other is key independence. Table 4.2 summarizes the results for analyzed algorithms. All of the analyzed algorithms, except for the Kronos, are able to fulfill the three requirements.

Table 4.2. Comparison of Security Considerations

	Forward Secrecy	Backward Secrecy	Key independence
IOLUS	Yes	Yes	Yes
Kronos	No	No	No
Hydra	Yes	Yes	Yes
GDH.2	Yes	Yes	Yes
GDH.3	Yes	Yes	Yes
BD	Yes	Yes	Yes
Octopus	Yes	Yes	Yes
CKA	Yes	Yes	Yes
HT-GKMA	Yes	Yes	Yes

Table 4.3 summarizes the general properties of each protocol with respect to IEEE 802.15.4's general requirements. The first aspect is multiple topology handling where the algorithms are analyzed to handle mesh, star and hybrid topologies stated in the standard. The second one is one of the basic limitations of IEEE 802.15.4, in which the algorithm is analyzed if it can handle both fully functional devices (FFDs) and reduced functionality devices (RFDs). Finally the last aspect in this table summarizes the additional limiting requirements of the algorithms.

The third comparison point is the efficiency of ACL list entries. Table 4.4 summarizes the results for the number of ACL list entries. When an algorithm does a data transformation while transmitting the message, it is uses the ACL list entries more efficiently since it cannot know the key of the end user or the whole group. Therefore, IOLUS-like algorithms are more advantageous. The other algorithms where all the group members use the same key, should keep the same group key for every member.

The next step is to compare the algorithms in terms performance. The cost of computations and the initial key distribution as well as the member addition and deletion costs of the different algorithms will be compared. However, only the comparisons between Diffie-Hellman based systems will be made since setting up a common com-

Table 4.3. Comparisons of Algorithms about Multiple Topology Handling and Device Support

	Multiple Topology Handling	**RFD and FFD Support**	**Additional Requirements for RFDs**
IOLUS	Yes	Yes	GSIs should be FFDs ; Large packet size in member leave may result in transmission inefficiency due to packet size constraints in IEEE 802.15.4
Kronos	Yes	Yes	Should use NTP
Hydra	No (Star topology is meaningless)	Yes	HS-Group Members should be FFDs
GDH.2	Yes	RFD support depends on the characteristics of the device (random number generation and exponentiation may be a problem)	Should generate random number
GDH.3	Yes	RFD support depends on the characteristics of the device (random number generation and exponentiation may be a problem)	Should generate random number
BD	Yes	RFD support depends on the characteristics of the device (random number generation and exponentiation may be a problem)	Should generate random number

	Multiple Topology Handling	**RFD and FFD Support**	**Additional Requirements for RFDs**
Octopus	Most hybrid topologies and star topology cannot be supported	Random number generation requirement for RFD may be problem	Leaders should be FFDs
CKA	Yes	Yes	RFDs to compute the key is costly
HT-GKMA	Yes	Yes	No

Table 4.4. Comparisons of Algorithms for ACL List Efficiency

	ACL List Entries
IOLUS	$\sim (r/n_{GSI} + n_{GSI}) \forall FFD; 1 \forall RFD$
Kronos	$(n + r) \forall FFD$ and RFD
Hydra	$(n + r) \forall FFD$ and RFD
GDH.2	$(n + r) \forall FFD$ and RFD
GDH.3	$(n + r) \forall FFD$ and RFD
BD	$(n + r) \forall FFD$ and RFD
Octopus	$(n + r) \forall FFD$ and RFD
CKA	$(n + r) \forall FFD$ and RFD
HT-GKMA	$n - 1 \forall FFD; 1 \forall RFD$

Table 4.5. Comparisons of Algorithms for Computational Efficiency

	Number of Exponentiations	Number of Rounds
GDH.2	$((n+r+3)(n+3))/2-1$	$n+2r$
GDH.3	$5(n+r)-6$	$n+2r+1$
BD	$n+r+1$	3
CKA	0 (other computational costs exists)	3
HT-GKMA	$5n-6$	$n+r/s$

putation base for all of the algorithms is impossible. Especially for the decentralized group key management algorithms, it is not possible to realize them in one single base example topology. Since every algorithm needs its custom components and their way of setups, computation cannot be done. Even when a setup is agreed, the algorithms used are not comparable with each other.

However, for the Diffie-Hellman based algorithms, a cost comparison can be made in terms of the number of exponentiations and the time needed to transmit a message and therefore, the number of rounds. The first row of Table 4.5 compares the number of exponentiations in which the total number of exponentiation operations needed to generate the group key is shown. The second row shows the number of rounds for each algorithm meaning that the number of steps which are taken for each member to generate the group key. According to these values, the time costs for each algorithm is computed in Table 4.6.

In Table 4.5 the total number of exponentiations is the sum of number of exponentiations made by FFDs and the number of exponentiations made by RFDs. For the number of rounds, RFDs' inability to send the messages directly is regarded and every number of RFD rounds is multiplied by two in order to reflect the fact that the message is transmitted first to the central FFD and then to the next group member. Using the values in Table 4.5 resulted in time calculations given in Table 4.6. According to Y. Amir [21], the average time cost for message sending varies between 0.75ms to 0.92 ms while the time cost for one exponentiation is 1.7 ms. The first one can be taken 0.83

Table 4.6. Time Performance Comparisons of Group Key Management Algorithms

	Approximate Time for Initial Key Establishment	Approximate Time for Member Addition	Approximate Time for Member Leave
GDH.2	$(n^2 + 7n + rn + 5r + 7)t/2$	$(n + r + 6)t$	$(n + r - 1/2)t$
GDH.3	$(11n/2 + 6r - 11/2)t$	$(n + r + 6)t$	$(n + r - 1/2)t$
BD	$(2n + 3r + 1)t$	$3t + (n + 2r + 1)t$	$(n + 2r + 2)t$
HT-GKMA	$(11n/2 - r/2s - 6)t$	$(r/sm + 1/2)t$ for RFDs; $(n + 5)t$ for FFDs	$(1 + r/sm)t$ for RFDs; $(n - 1/2)t$ for FFDs

ms in average, making up approximately half of the exponentiation cost. Therefore, in Table 4.6 time cost for explanation is taken as t and time cost for message sending is taken as $t/2$.

The calculations for GDH.2 and GDH.3 in Table 4.6 are straightforward. It is the sum of values in Table 4.5. For BD, however, the number of rounds is stated as 3 in Table 4.5, but it is not taken as the communication cost value in Table 4.6. The reason is that in the first two rounds in BD, a broadcast message is sent by every member. This value is the the value that has been taken into consideration while computing the time cost for communication. For RFDs, this value is doubled because the message should be first sent to its FFD.

Using the outputs in Table 4.6 and the value 1.7 ms for t, results in a series of graphs where the initial key distribution times, the member addition times and the member deletion times of the GDH.2, GDH.3, BD and HT-GKMA algorithms are compared. Here, s (number of star subtopologies), is taken as 1 in order to picture the worst case scenario, and m is taken as 7.

Figures 4.6 and 4.7 show the results when the number of FFDs are constant and

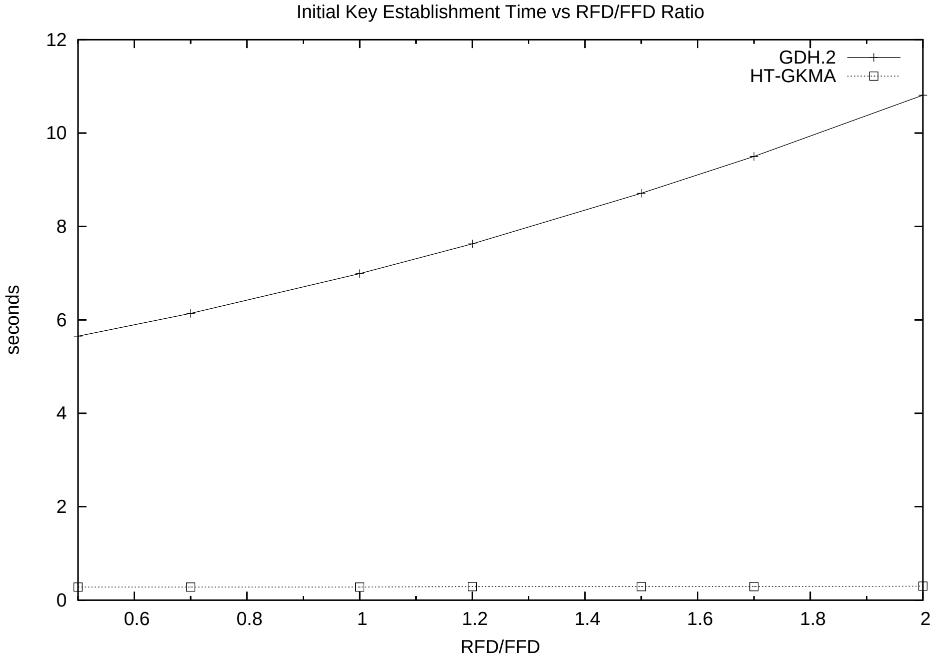


Figure 4.6. Initial key establishment time versus RFD/FFD ratio

equal to thirty but the RFD/FFD ratio is variable. Therefore, if the RFD/FFD ratio is equal to 0.5, there are 15 RFDs and 30 FFDs in the group communication network. Here, the graphs are split into two, in order to show the results in more detail. Figure 4.6 shows the results for GDH.2 and HT-GKMA, while the other shows the results for GDH.3, BD and HT-GKMA algorithms. It can be seen that GDH.2 gives the highest time cost in initial key establishment. The most efficient algorithm is the HT-GKMA when the RFD/FFD ratio is greater than 1,2.

It should be noted that while making the computations, exponentiation has been taken as the major cost. But in BD, there is a hidden cost in step three. In the last step, while computing the key, multiplications should be made. It should be noted that, multiplications can be a major cost when the number of members increases. This would increase the overall cost of the BD algorithm.

The second comparisons are made by the changing number of FFDs with the constant RFD/FFD ratio equals to two. That is, when the number of FFDs is equal to 5, there exist 10 RFDs in the group communication network. For the initial key

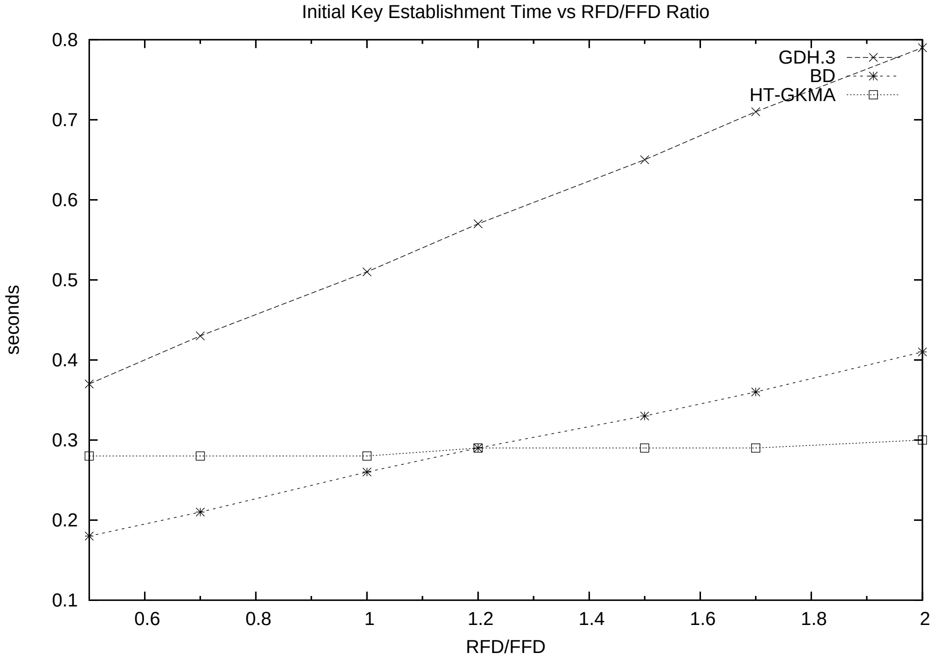


Figure 4.7. Initial key establishment time versus RFD/FFD ratio

establishment, two different graphs in Figure 4.8 and Figure 4.9 show GDH.2 and GDH.3, BD algorithms respectively.

The second time cost comparison subject is the member addition time which is also analyzed using two scenarios. In the first scenario, the number of FFDs is constant(30) and the RFD/FFD ratio is changing. Conversely, in the second scenario the RFD/FFD ratio is constant and equal to two, while the number of FFDs varies between 5 and 35. Figures 4.10 and 4.11 shows the results respectively. Here, GDH.2 and GDH.3 yields the same results while BD has the worst time cost for member addition. HT-GKMA has the lowest results.

The final timing cost comparisons are done for the member deletion operation. The two scenarios used in the previous comparisons are also applied exactly in this step. The results are shown in Figures 4.12 and 4.13. GDH.2 and GDH.3 yield the same results while BD has the worst time-cost for member leave. HT-GKMA has the lowest results amongst all.

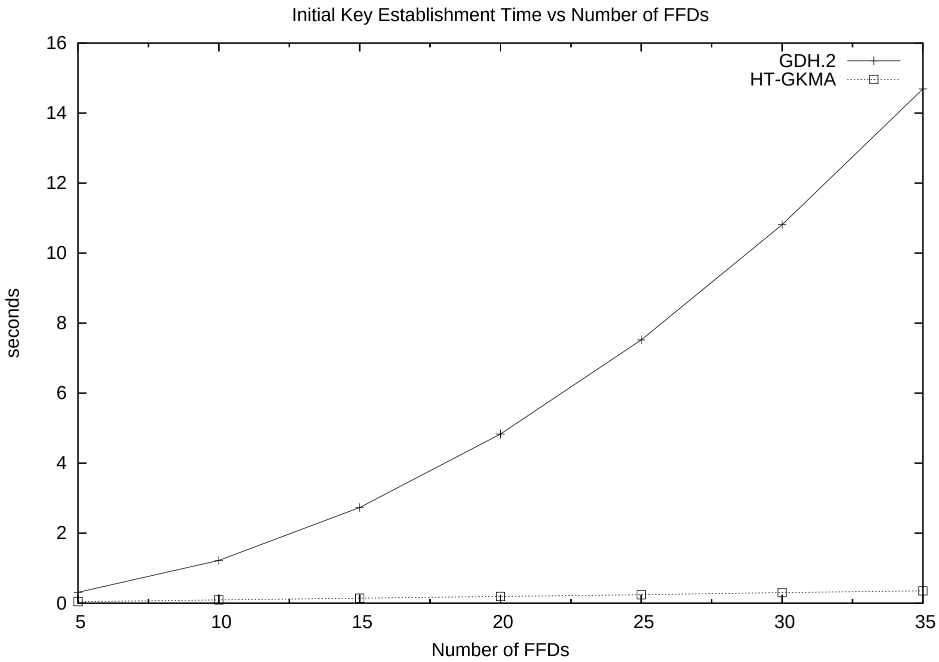


Figure 4.8. Initial key establishment time versus number of FFDs

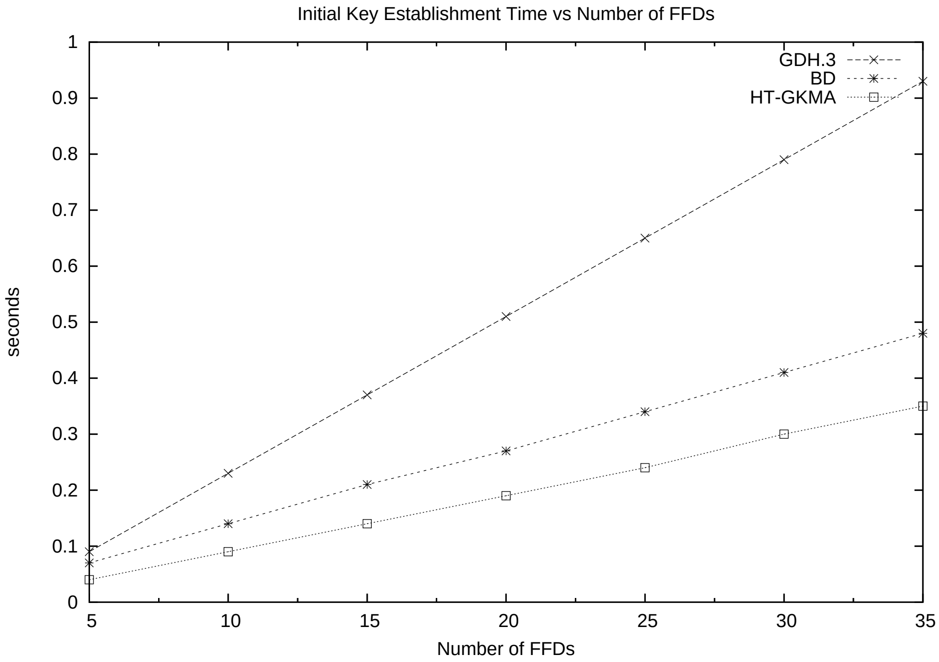


Figure 4.9. Initial key establishment time versus number of FFDs

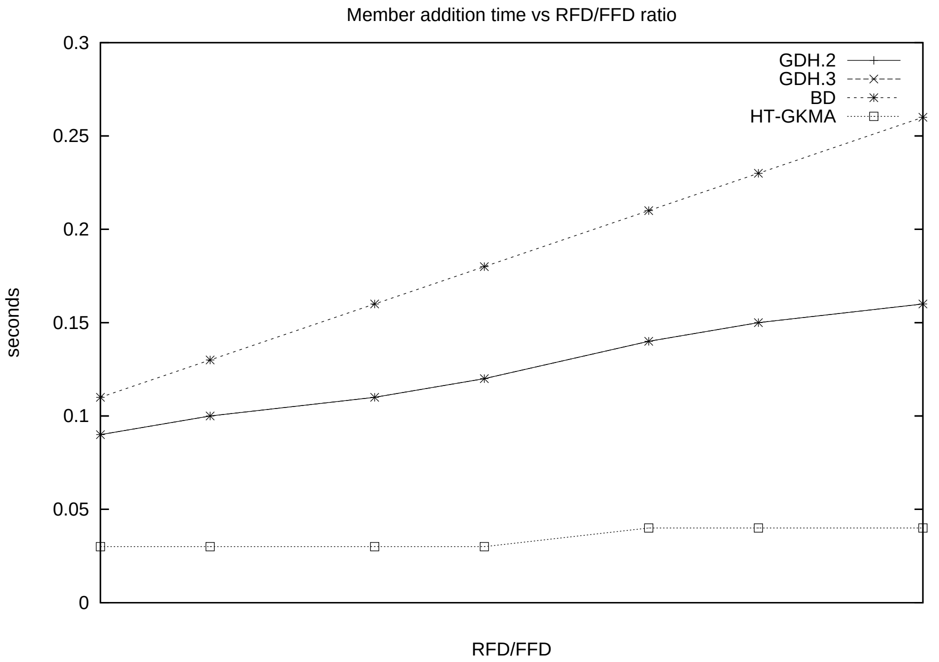


Figure 4.10. Member addition time versus RFD/FFD ratio

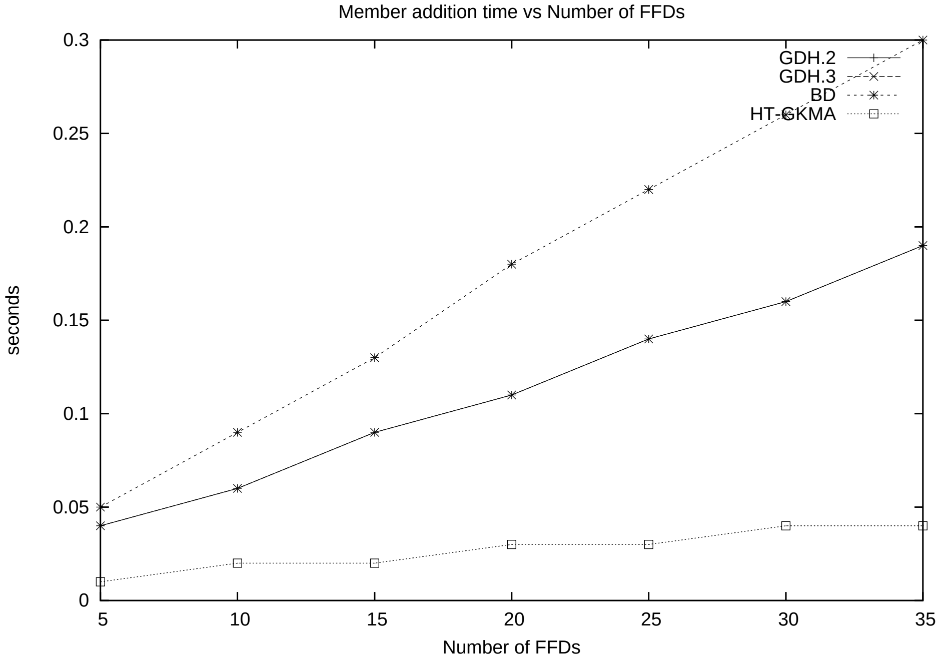


Figure 4.11. Member addition time versus number of FFDs

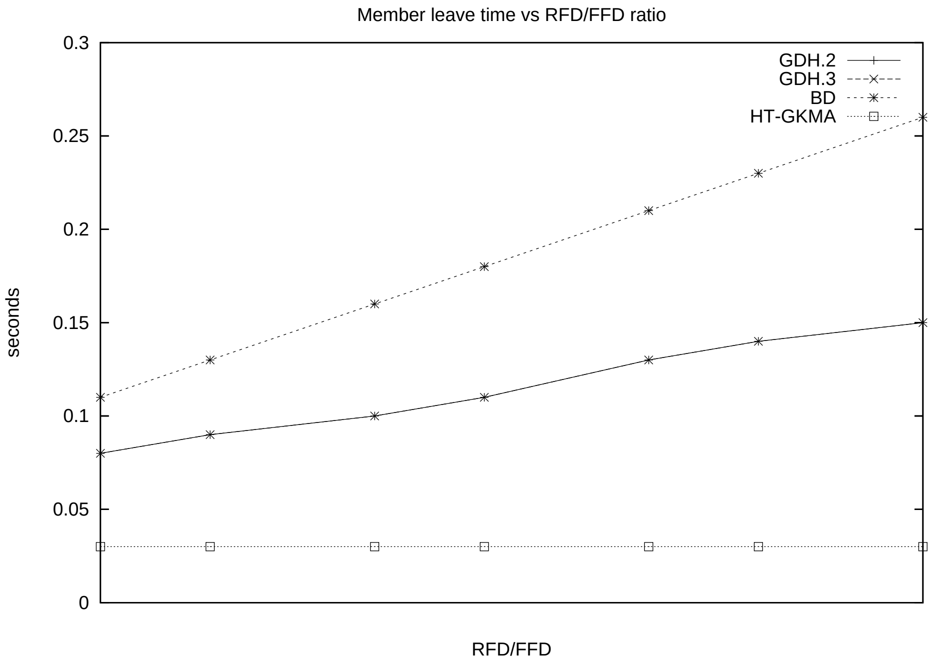


Figure 4.12. Member leaving time versus RFD/FFD ratio

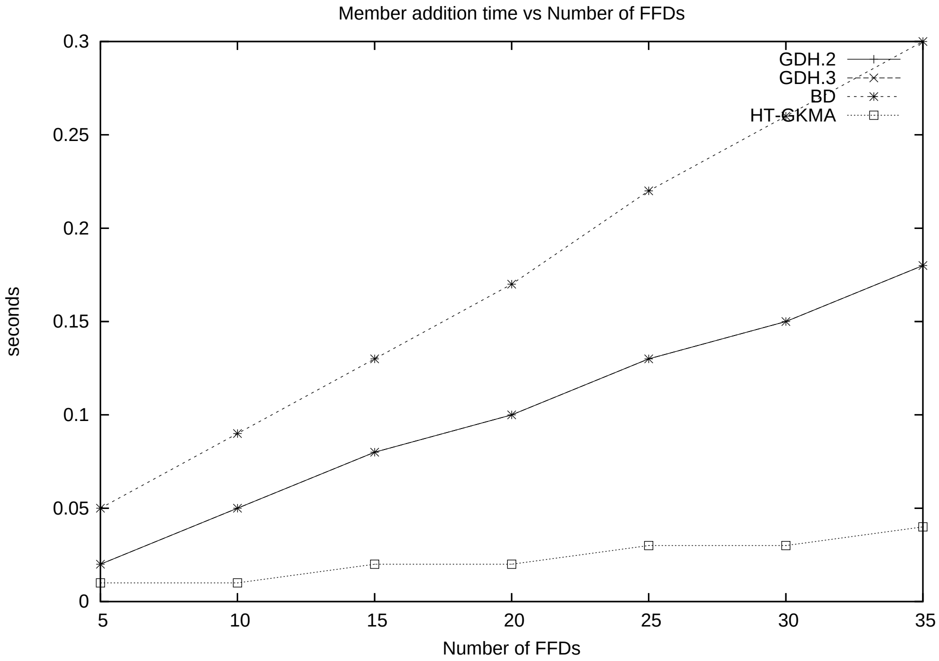


Figure 4.13. Member leaving time versus number of FFDs

5. CONCLUSIONS

In an IEEE 802.15.4 network, distribution of the group keys for group communication to ensure group communication security, as well as the efficiency of the member addition and deletion algorithms make up some of the security problems in IEEE 802.15.4 [15]. Group key management algorithms such as Iolus, Hydra, Kronos, GDH.2, GDH.3, Octopus, BD, CKA have already been proposed for different reference environments. Each algorithm has its pros and cons when used in an IEEE 802.15.4 environment and the main limitation is the IEEE 802.15.4's ability to form networks with different topologies and different kinds of IEEE 802.15.4 devices, that is, FFDs and RFDs.

When a network is formed using only FFDs and the topology is mesh, such a network is in fact equivalent to an ad-hoc wireless network in which each node has the message routing capability of any other node. In this kind of topology, distributed group key management algorithms ensure flexibility, therefore they are more preferable than centralized and decentralized group key management algorithms. Among the distributed group key management algorithms, those in which all the group members contribute to the group key generation are more fault tolerant and diminish the risks of vicious key generation by a single entity.

However, when the network topology is star and the devices are RFDs, distributed approaches are not efficient anymore because of limitations of the star topology. For a star topology network, key management can be effectively done using either centralized or decentralized group key management approaches. Since there exists a central management point for communication, that point can also handle the key management of group communication. But, when a centralized approach is used, in the case of member addition and deletion all of the star topology members should change the keys. Therefore, a local rekeying approach would be useful especially for the large star topology IEEE 802.15.4 networks.

Following a local rekeying approach leads us to a conclusion where a hybrid solution would be much more suitable for an IEEE 802.15.4 network, where the mesh connected FFDs use a distributed algorithm while the star connected RFDs' group key management is done by a decentralized algorithm that uses local rekeying. This way, every possibility of forming an IEEE 802.15.4 is covered keeping in mind that hybrid networks with both star and mesh topologies can also be setup.

Forming a hybrid approach and covering every possibility adds a overhead to FFDs. They should keep two different algorithms in mind. Assuming that the FFDs are more capable devices, this approach lowers the computation complexity on RFDs. Shifting all the computation and transformations to FFDs help RFDs be more energy efficient.

The proposed group key management scheme, HT-GKMA (Hybrid topology group key management algorithm), is especially designed for IEEE 802.15.4 keeping in mind the standard's requirements. Then, the newly proposed hybrid group key management scheme is compared with existing group key management algorithms to demonstrate whether it is functioning as estimated in an IEEE 802.15.4 network environment. The first comparison aspect is the security of the proposed algorithms. This includes forward and backward key secrecy, as well as key independence. All of the analyzed algorithms, except for the Kronos, are able to fulfill the three requirements. The next comparison is about the general properties of each protocol with respect to IEEE 802.15.4's general requirements. The first aspect in here is multiple topology handling where the algorithms are analyzed to handle mesh, star and hybrid topologies stated in the standard. The second one is about one of the basic limitations of IEEE 802.15.4, in which the algorithm is analyzed if it can handle both fully functional devices (FFDs) and reduced functionality devices (RFDs). Finally, a summary of the additional limiting requirements of the algorithm is also discussed. Previously proposed group key management algorithms such as GDH.2, GDH.3, BD, Octopus could not handle RFDs. Furthermore, Hydra and Octopus group key management algorithms could not handle both star and mesh topologies. The next comparison is about the ACL list efficiency of the group key management algorithms. Local rekeying based group key management approaches gave smaller ACL lists as a result. Finally, time cost comparisons of initial key establishment, member addition and leave between the algorithms are carried out between the Diffie-Hellman based algorithms, including HT-GKMA, which gave the best results amongst other algorithms such as Burmester-Desmedt, GDH.2 and GDH.3.

Comparisons of HT-GKMA with various group key management algorithms showed that the most key distribution /redistribution time effective group key management algorithm for IEEE 802.15.4 based networks that supports different capability devices and topologies is the proposed algorithm, HT-GKMA. HT-GKMA can be used as the group key management algorithm in IEEE 802.15.4 networks where security is needed for multicasting environment.

In this thesis, a new group key management algorithm is proposed to use with IEEE 802.15.4 standard. In future work, the implementation of this algorithm in an IEEE 802.15.4 network can be examined in order to compare the theoretical conclusions with real-life implementation.

REFERENCES

1. IEEE Standard for Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks– Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs) http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf

2. Rafaeli, S. and D. Hutchison, *A survey of Key Management for Secure Group Communication*, ACM Computing Surveys, Vol 35, No.3, September 2003, pp. 309-329

3. Challal, Y. and H. Seba, *Group Key Management Protocols: A Novel Taxonomy*, International Journal of Information Technology, Vol 2, No: 1 2005, ISSN:1305-2403

4. Mittra, S., *IOLUS: A Framework for Scalable Secure Multicasting*, Proceedings of the ACM SIGCOMM '97, September 14-18, 1997

5. Rafaeli, S. and D. Hutchison, *Hydra: A Decentralized Group Key Management*, Proceedings of the 11th IEEE WETICE Workshop, 2002

6. Rafaeli, S., L. Mathy, and D. Hutchison *EHBT: An efficient protocol for group key management*,Third International Workshop on Networked Group Communications, volume LNCS, London, UK, Nov 2001. Springer-Verlag.

7. Setia, S., S. Koussih, and S. Jajodia,*Kronos: A Scalable Group Re-keying Approach for Secure Multicast*,2000 IEEE Symposium on Security and Privacy, Oakland CA, May 2000.

8. Mills, D.L.,*Network Time Protocol (version 3) Specification and Implementation*, RFC1305, March 1992.

9. Harney, H. and E. Harder, *Logical Key Hierarchy Protocol Internet Draft*, draft-

harney-sparta-lkhp-sec-00.txt, March 1999.

10. Wallner, D., E. Harder and R. Agee, *Key Management for Multicast: Issues and Architectures*, RFC 2627, June 1999.

11. Steiner, M., G. Tsudik and M. Waidner, *Diffie-Hellman Key Distribution Extended to Group Communication*, ACM Conference on Computer and Communication Security, pp 31-37, March 1996

12. Steiner, M., G. Tsudik and M. Waidner, *CLIQUES: A New Approach to Group Key Agreement*, IEEE International Conference on Distributed Computing Systems, May 1998

13. Ateinese, G., M. Steiner and G. Tsudik, *Authenticated Group Key Agreement and Friends*, Proceedings of the 5th ACM Conference on Computer and Communication Security, Nov 1998

14. Eschenauer, L. and V. Gligor, *A key-management scheme for distributed sensor networks*, Conference on Computer and Com-munications Security. Proceedings of the 9th ACM conference on Computer and communications security, Washington, DC, 2002

15. Sastry, N. and D.Wagner, *Security Considerations for IEEE 802.15.4 networks*, WISE'04, Oct 2004

16. Zigbee Alliance, *http://www.zigbee.org*

17. Harney, H. and C. Muckenhirn *Group Key Management Protocol (GKMP) Architecture*, rfc 2094,July 1997

18. Anton, E.R and O.C.M.B. Duarte, *Group Key Establishment in Wireless Ad Hoc Networks*, Workshop em Qualidade de Serviço e Mobilidade, 2002

19. Wallner, D.M, E.G. Harder and R.C. Agee, *Key Management for Multicast: Issues*

and Architecture, Internet Draft, draft-wallner-key-arch-01.txt, September 1998

20. Shnayder, V., B. Chen and K. Lorincz, *Sensor Networks for Medical Care*, Technical Report TR-08-05, Division of Engineering and Applied Sciences, Harvard University, 2005.

21. Amir, Y., Y. Kim, C. Nita-Rotaru and G. Tsudik, *On the Performance of Group Key Agreement Protocols*,Proceedings of the 22nd IEEE International Conference on Distributed Computing Systems, June 2002

22. Burmester, M. and Y. Desmedt, *A Secure And Efficient Conference Key Distribution System*, In Advances in Cryptology, EUROCRYPT '94, Lecture Notes in Computer Science. Springer-Verlag, Berlin Germany, 1994

23. Becker, C. and U. WILLE, *Communication Complexity of Group Key Distribution*, Proceedings of the 5th ACM Conference on Computer and Communications Security. (San Francisco, Calif., Nov.). ACM, 1998

24. Boyd, C., *On Key Agreement and Conference Key Agreement*, Proceedings of the Information Security and Privacy: Australasian Conference. Lecture Notes in Computer Science, vol. 1270. Springer-Verlag, 294302,1997