Spring 5-4-2011

Attacking and Securing Beacon-Enabled 802.15.4 Networks

Sang Shin Jung

ATTACKING AND SECURING BEACON-ENABLED 802.15.4 NETWORKS

by

SANG SHIN JUNG

Under the Direction of Dr. Raheem Beyah

ABSTRACT

The IEEE 802.15.4 has attracted time-critical applications in wireless sensor networks (WSNs) because of its beacon-enabled mode and guaranteed timeslots (GTSs). However, the GTS scheme's security still leave the 802.15.4 MAC vulnerable to attacks. Further, the existing techniques in the literature for securing 802.15.4 either focus on non beacon-enabled 802.15.4 or cannot defend against insider attacks for beacon-enabled 802.15.4. In this thesis, we illustrate this by demonstrating attacks on the availability and integrity of the beacon-enabled 802.15.4. To proof the attacks, we implement the attacks using Tmote Sky motes for a malicious node along with regular nodes. We show that the malicious node can freely exploit the beacon frames to compromise the integrity and availability of the network. For the defense, we present beacon-enabled MiniSec (BCN-MiniSec) and analyze its cost.

INDEX WORDS: Insider attacks, Beacon-enabled 802.15.4, Wireless sensor networks, MAC misbehavior

ATTACKING AND SECURING BEACON-ENABLED 802.15.4 NETWORKS

by

SANG SHIN JUNG

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of
Master of Science
in the College of Arts and Sciences
Georgia State University

2011

ATTACKING AND SECURING BEACON-ENABLED 802.15.4 NETWORKS

by

SANG SHIN JUNG

Chair:	Dr. Raheem Beyah
Committee:	Dr. Anu Bourgeois
	Dr. WenZhan Song

Electronic Version Approved:

Office of Graduate Studies

College of Arts and Sciences

Georgia State University

May 2011

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Raheem Beyah, for all of his guidance and support.
I would also like to thank Dr. Anu Bourgeois and Dr. WenZhan Song.
I also want to express my gratitude to the Communication Assurance and Performance CAP Group, especially to Marco Valero, for all the support and ideas. I also extend my gratitude to Dr. Rajshekhar Sunderraman for all of his support.
Finally, I would like to thank especially my wife, Kyongah, for all of her support and my family and friends.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

Wireless sensor networks (WSNs) have emerged quickly and attracted a number of diverse applications. The use of these applications ranges from residential to government. For example, AlertMe home monitoring [1] is a residential system that enables secure indoor and outdoor home environment monitoring with simple contact and passive infrared (PIR) sensors. If AlertMe detects intruders, it immediately reports the intrusion to the homeowner. The military is also using WSNs to detect an adversary's behavior and location. For example, seismic sensors can be used to detect the movement of heavy artillery (e.g., tanks) in the battlefield. In either case, not receiving information about the environment in a time-sensitive manner can have significant consequences. To provide support for time-sensitive communication, the IEEE 802.15.4 standard provides a beacon-enabled mode. Unlike non beacon-enabled mode, the beacon-enabled mode in a 802.15.4 network employs a few end device nodes and a centralized node (i.e., personal area network (PAN) coordinator) which broadcasts beacons to synchronize the nodes in the network, manages guaranteed timeslots (GTS) (de)allocation requests from the nodes, and assigns dedicated slots for transmissions of the nodes through beacons. The beacon broadcast and GTS management scheme are the most critical parts of real-time delivery of time-sensitive data during the contention free period (CFP) [2–7]. Many researchers have focused on improving the performance or energy efficiency of beacon-enabled 802.15.4 MAC and the use of its GTS scheme. For example, the IPP-HURRY research group has analyzed the delay bound of GTS allocations to maximize the throughput of each GTS allocation for real-time sensor networks [3, 4]. In addition, in [5] the authors present a case study of Siemens Industry Automation Division that requires real-time delivery of short alarms/messages. The case study evaluates GTS allocation to maximize low latency of its scheme. Although there has been a significant emphasis on improving the performance of beacon-enabled 802.15.4 networks, there has been little work on securing them. This is significant, given that the GTS management scheme of the PAN coordinator does not verify the ID of each node that requests GTSs. Further the nodes in the network do not validate the PAN coordinator that broadcasts beacons. Therefore, an inside attacker can easily compromise the guaranteed data transmissions from the time-sensitive applications in the beacon-enabled 802.15.4 network by either impersonating legitimate nodes (existing in the PAN or not) or the PAN coordinator (e.g., implement a Sybil attack [8] at the MAC layer).

In this thesis, we demonstrate six attacks that are possible by an inside attacker in a beacon-enabled 802.15.4 network. The inside attacker targets the vulnerabilities of the beacon broadcast and the GTS management scheme. The contributions of this thesis include the discovery of vulnerable properties of the beacon-enabled mode in the IEEE 802.15.4 standard and the implementation and analysis of six potential insider attacks associated with those vulnerabilities. We also present an extension of MiniSec [9], beacon-enabled MiniSec (BCN-MiniSec), to defend against these attacks and examine its cost.

The rest of this thesis is organized as follows. We review some related works including several security protocols for WSNs and attacks on beacon-enabled IEEE 802.15.4 in Chapter 2. In Chapter 3, we briefly illustrate the beacon broadcast and the GTS management scheme and explain their vulnerabilities. In Chapter 4, we present the experiment design and show the hardware and software components. In Chapter 5, we first define an attack model and present an overview of the six attacks against the vulnerabilities. In Chapter 6, we describe the implementation of the attacks. In Chapter 7, we show the analysis of each attack's results based on the captured data. We present BCN-MiniSec to defend against these attacks in Chapter 8 and conclude our work in Chapter 9.

Chapter 2

RELATED WORK

In this chapter we categorize current 802.15.4 defense mechanisms into non beacon-enabled mode and beacon-enabled mode according to the literature and highlight their limitations. We also discuss the difference between our attacks on beacon-enabled 802.15.4 networks and others previously demonstrated.

2.1 Defense Mechanisms in Beacon-Less Mode

In [10, 11], the authors propose using the received signal strength indication (RSSI) to identify nodes conducting a Sybil attack. The basic idea of RSSI-based methods is that sensor nodes at different locations can be differentiated by the different RSSIs. In [10], M. Demirbas et al. calculate the ratio of RSSIs to improve traditional RSSI-based solutions. In [11], J. Yang et al. propose K-means cluster analysis that can be applied to RSSI readings. However, RSSI-based solutions can be evaded by malicious nodes with mobility. Another approach to securing beacon-less 802.15.4 networks focuses on the use of cryptography. In [12] the authors propose light-weight identity certificates to distinguish between legitimate nodes and malicious nodes using multiple stolen or forged IDs, while the authors of [13–16] focus on key distribution and management algorithms to provide this protection. However, it is not practical for resource constrained sensor devices to use highly expensive key distribution methods. Link layer security protocols constitute another category of defense mechanisms for beacon-less 802.15.4 networks. SPINS, TinySec, and MiniSec [9, 17, 18] fall in this category and are designed specifically for energy constrained sensor nodes and provide data authentication and confidentiality in the link layer. However, these protocols are susceptible to failures when a malicious node in the network (e.g., a compromised node or a malicious insider) acquires a shared pair-wise key or a network-wide secret key. Moreover, even if their shortcomings are excluded, none of the aforementioned schemes can be directly applied to beacon-enabled 802.15.4 networks. This is because in addition to the data protection provided by the aforementioned schemes, beacon-enabled mode control messages (e.g., beacon broadcasts from the PAN coordinator) must also be secured, and as pointed out by Perrig et al. in [17], traditional data authentication techniques cannot be used to provide broadcast beacon authentication.

2.2 Defense Mechanisms in Beacon-Enabled Mode

Few defense methods have been proposed for beacon-enabled mode. One RSSI-based solution for beacon-enabled mode was proposed by F. Amini et al. in [19]. The authors proposed an RSSI solution where they introduced the use of a disc number and a device ID. However, if a malicious node is close enough to a legitimate node in the same PAN (i.e., an inside attacker), its RSSI may be confused with the RSSI of the legitimate node. The IEEE 802.15.4 standard [20] also has built-in security mechanisms to provide data confidentiality and data authenticity. However, in [21], N. Sastry et al. point out that these security mechanisms have vulnerabilities related to the initialization vector (IV) management, key management, and integrity protection. Moreover, the security mechanism only guarantees data authentication, not authentication for beacon broadcasts. Alim et al. introduce EAP-Sens in [22], which provides entity authentication and key management to validate each device ID with the extensible authentication protocol (EAP) [23] using EAP-generalized pre-shared keys (EAP-GPSKs) [24]. However, EAP-Sens uses the built-in security mechanisms of the 802.15.4 standard to secure the communication between nodes and the PAN coordinator, which means that it has the same problems as the security mechanisms in the 802.15.4 standard. Overall, neither the aforementioned detection mechanisms nor secure link layer protocols for the beacon-enabled mode are effective in the case of inside attackers.

2.3 Attacks on Beacon-Enabled 802.15.4 Networks

In [25], R. Sokullu et al. use ns-2 simulations to demonstrate GTS attacks on the 802.15.4 MAC, particularly in beacon-enabled mode. The GTS attacks were divided into four different scenarios: One Intelligent Attacker (OIA), One Random Attacker (ORA), Two Intelligent Attackers (TIAs), and Two Random Attackers (TRAs). Both the OIA and TIAs scenarios target the maximum number of GTS slots assigned to one legitimate node. In contrast, the ORA and TRAs scenarios attack just one randomly chosen GTS. The main goal of the GTS attacks in [25] is to create collisions during the CFP to deny the use of GTSs. In contrast, the six attacks that we present seek to exploit the beacon-enabled 802.15.4 MAC by inducing scenarios of unfairness and exhaustion [26, 27].

In addition to presenting different types of attacks compared to those discussed in [25], we implemented our attacks on real devices (i.e., Tmote Sky motes) rather than in simulation. This latter point is extremely important for 802.15.4 MAC layer attacks, because in addition to the challenge of accurately modeling physical layer interference, simulations do not take into account constraints imposed by the hardware, operating system, and applications, which can lead to simplified attack scenarios. This is especially pronounced in resource-constrained devices (e.g., Tmote Sky motes). For example, to implement the Sybil attack (at the MAC layer) in TinyOS, we modified the timer function of TinyOS (in TimerC.nc) to make it multithreaded so each fake node could use an instance. Each instance now has to compete internally (within TinyOS) to gain access to the node's resources (e.g., processor, transceiver), making this attack much more difficult to conduct. This small, but noticeable nuance is not present in simulation tools.

In [28] we introduced several attacks on the beacon-enabled 802.15.4 network. This work extends [28] with the addition of 3 new implemented attacks as well as the presentation of BCN-MiniSec to defend against the attacks.

Chapter 3

PROBLEM STATEMENT

In this chapter, we briefly explain the beacon broadcast and the GTS management scheme of the IEEE 802.15.4 standard. Additionally, we state the vulnerabilities of these schemes.

3.1 Beacon Broadcasts

The IEEE 802.15.4 standard [20] operating in beacon-enabled mode defines the superframe (SF) that consists of a contention access period (CAP), a contention free period (CFP), and an inactive period as shown in Figure 3.1. The active period of the CAP and the CFP is divided into 16 timeslots during which the nodes in the network should synchronize with and transmit data. The timeslots can be synchronized through beacons that the personal area network (PAN) coordinator periodically transmits at intervals defined by the *macBeaconOrder* value. Upon receiving the beacons, the nodes take the beacon order (BO) and SF order (SO) from the SF specification field in Figure 3.2 (b) and synchronize the timeslot interval, SF duration (SD), and beacon interval (BI) to the SF of the PAN in Figure 3.1.

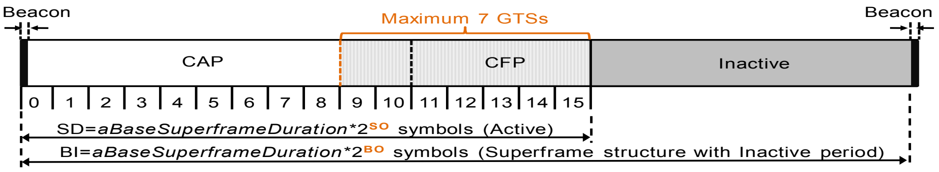

Beacon
Maximum 7 GTSs
Beacon
CAP
CFP
Inactive
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
SD=*aBaseSuperframeDuration**2^{SO} symbols (Active)
BI=*aBaseSuperframeDuration**2^{BO} symbols (Superframe structure with Inactive period)

Figure 3.1. The superframe structure of beacon-enabled 802.15.4.

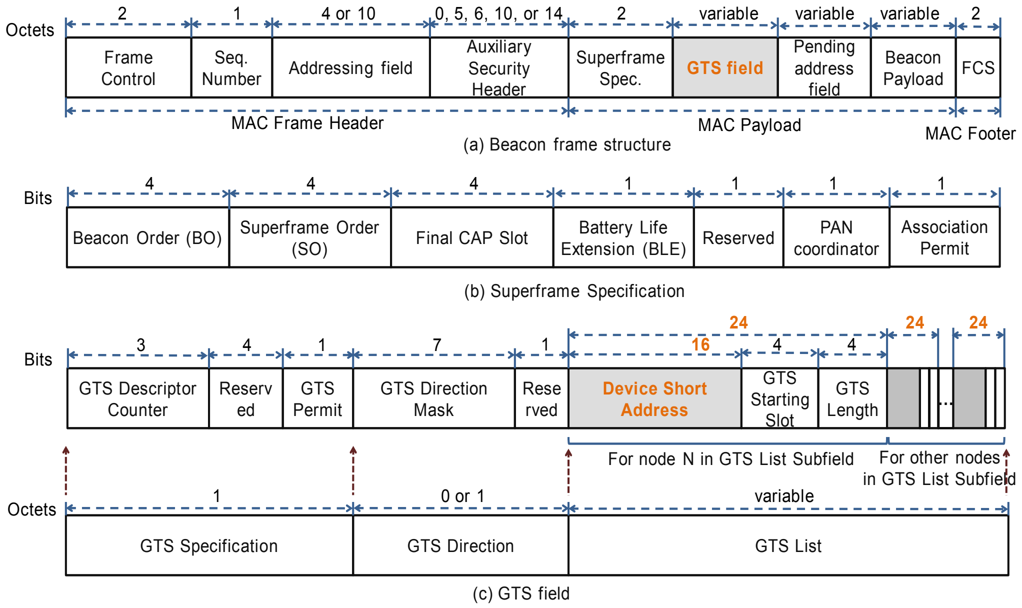


Figure 3.2. The beacon frame structure of the beacon-enabled 802.15.4 in detail: (a) beacon frame structure (b) SF specification structure, and (b) GTS field structure in beacon frame.

3.2 Vulnerability of Beacon Broadcasts

Verification of the PAN coordinator: The two important values, BO and SO, can cause the nodes to change their internal timers used for synchronization and transmitting messages. However, when processing the beacons received, legitimate nodes do not authenticate the beacons and cannot tell whether they really came from the PAN coordinator. The nodes only confirm that the PAN ID in the packet is the same as the value used during bootstrapping of the network. Thus, if a malicious node sends beacons with the same PAN ID, the nodes process the malicious beacons the same as those from the PAN coordinator as shown in Figure 3.3. As mentioned in Chapter 2, the data authentication of the standard does not apply to beacon broadcasts.

3.3 GTS Management Scheme

The beacon frames contain the guaranteed timeslots (GTSs) information and directions used by nodes to transmit data during the CFP. The structure of the beacon frame and the

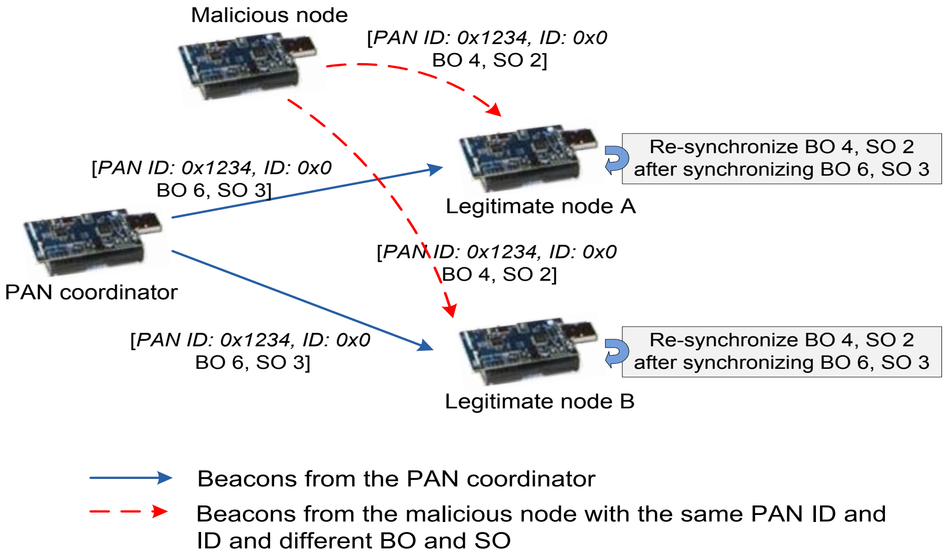

Malicious node
[PAN ID: 0x1234, ID: 0x0
BO 4, SO 2]
Re-synchronize BO 4, SO 2
after synchronizing BO 6, SO 3
Legitimate node A
[PAN ID: 0x1234, ID: 0x0
BO 6, SO 3]
PAN coordinator
[PAN ID: 0x1234, ID: 0x0
BO 4, SO 2]
[PAN ID: 0x1234, ID: 0x0
BO 6, SO 3]
Re-synchronize BO 4, SO 2
after synchronizing BO 6, SO 3
Legitimate node B
Beacons from the PAN coordinator
Beacons from the malicious node with the same PAN ID and ID and different BO and SO

Figure 3.3. A malicious node impersonating the PAN coordinator and broadcasting false BO and SO with the same PAN ID and the PAN coordinator's ID.

GTS field are shown in Figure 3.2 (a) and (c) respectively.

As shown in Figure 3.1, the PAN coordinator defines that each SF can have a maximum of seven GTSs for the CFP other than $aMinCAPLength$ in [20]. The GTSs must be assigned to legitimate nodes issuing GTS allocation requests to the PAN coordinator. Then, the assigned slots should be released by the PAN coordinator after receiving a GTS deallocation request from the same legitimate node. We briefly explain the normal GTS allocation and deallocation processes below.

GTS Allocation: If a legitimate node has data to transmit, it generates a GTS allocation request. The PAN coordinator will allocate an available GTS to the legitimate node, and all subsequent beacon frames will contain the GTS descriptor defining the device address, GTS slot and direction. Upon receiving the beacon with the GTS descriptor, the legitimate node will schedule the pending packet to be transmitted at the allocated GTS. The GTS allocation process is shown in Figure 3.4 (a).

GTS Deallocation: The GTS deallocation occurs after the GTS descriptor has been transmitted for $aGTSDescPersistenceTime$ beacons by the PAN coordinator or when the legitimate node using the GTS sends an explicit GTS deallocation request. The GTS deallocation process is shown in Figure 3.4 (b).

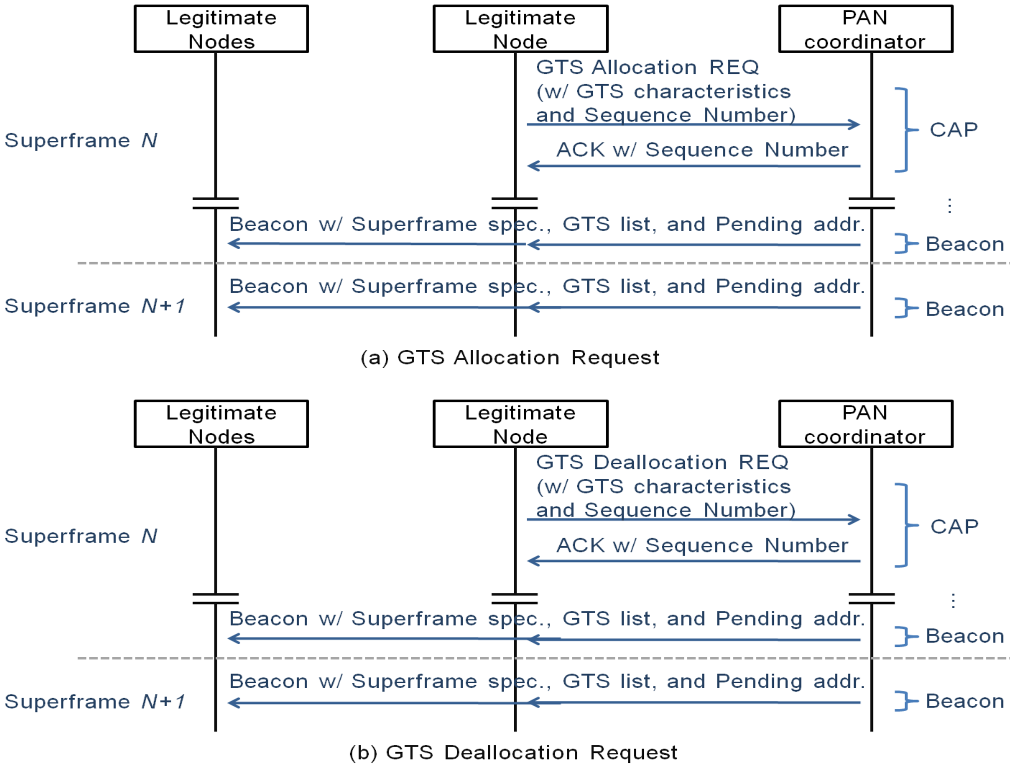

Legitimate Nodes
Legitimate Node
PAN coordinator
GTS Allocation REQ (w/ GTS characteristics and Sequence Number)
ACK w/ Sequence Number
CAP
Superframe N
Beacon w/ Superframe spec., GTS list, and Pending addr.
Beacon
Superframe N+1
Beacon w/ Superframe spec., GTS list, and Pending addr.
Beacon
(a) GTS Allocation Request
Legitimate Nodes
Legitimate Node
PAN coordinator
GTS Deallocation REQ (w/ GTS characteristics and Sequence Number)
ACK w/ Sequence Number
CAP
Superframe N
Beacon w/ Superframe spec., GTS list, and Pending addr.
Beacon
Superframe N+1
Beacon w/ Superframe spec., GTS list, and Pending addr.
Beacon
(b) GTS Deallocation Request

Figure 3.4. GTS allocation and deallocation procedure.

3.4 Vulnerabilities of GTS Management Scheme

The PAN coordinator manages a list of GTSs to control the network access during the CFP. However, the GTS management scheme has the following vulnerabilities.

CAP Maintenance: According to the IEEE 802.15.4 standard, the PAN coordinator can perform several preventative actions to keep $aMinCAPLength$. One of these actions is to deallocate unused GTSs within every $2{*}n$ SFs, where n is defined as either $2^{(8-macBeaconOrder)}$ $(0 \leq macBeaconOrder \leq 8)$ or $(9 \leq macBeaconOrder \leq 14)$. However, if a malicious node keeps constantly sending either GTS requests or data at the assigned GTSs during the CFP, the preventative action is ineffective.

Verification of Sensor Nodes' IDs: In the 802.15.4 GTS management scheme, the PAN coordinator manages the identities of legitimate nodes requesting one or more GTSs. The PAN coordinator assigns GTSs to the nodes, deallocates the assigned slots, and avoids duplicated GTS requests from the same legitimate node. However, as shown in Figure 3.5 the PAN coordinator only checks the sensor nodes' IDs (a short 2-octet address) and the sequence number of the packets. Thus, a malicious node can easily evade the verification process for sensor nodes' IDs by using new forged IDs or impersonating legitimate nodes in the network.

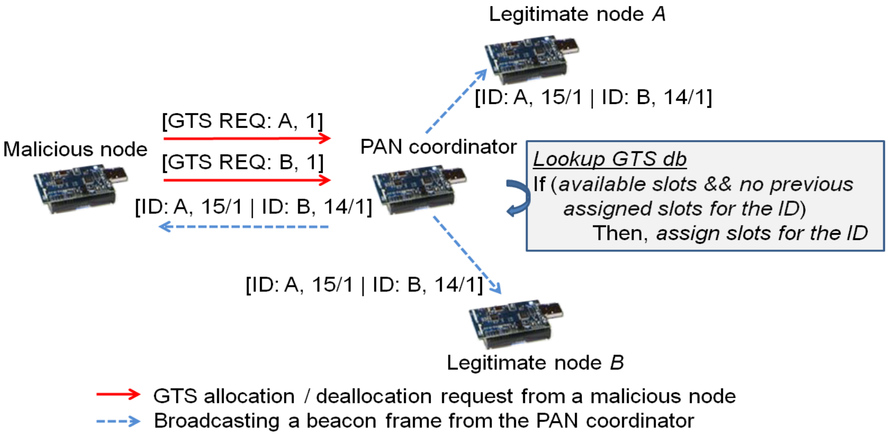

Legitimate node *A*
[ID: A, 15/1 | ID: B, 14/1]
[GTS REQ: A, 1]
Malicious node
[GTS REQ: B, 1]
PAN coordinator
[ID: A, 15/1 | ID: B, 14/1]
Lookup GTS db
If (*available slots && no previous assigned slots for the ID*)
Then, *assign slots for the ID*
[ID: A, 15/1 | ID: B, 14/1]
Legitimate node *B*
GTS allocation / deallocation request from a malicious node
Broadcasting a beacon frame from the PAN coordinator

Figure 3.5. A malicious node impersonating legitimate nodes A and B IDs.

Chapter 4

EXPERIMENT DESIGN

In this chapter, we present a network design and hardware and software components used for the experiments and implementation.

4.1 Network Design

In this thesis, we deploy wireless sensor nodes supporting the IEEE 802.15.4 standard and its beacon-enabled mode. In general, the beacon-enabled 802.15.4 network consists of few groups of clusters. One cluster can be composed of one PAN coordinator and few nodes. For the experiments, we arrange a small cluster that consists of one PAN coordinator and three nodes including a malicious node. The PAN coordinator broadcasts beacons and receives sensed data from the nodes. The nodes sense the temperature and humidity around the experiment area and transmit the data to the PAN coordinator during the CAP or the CFP. The nodes do not communicate with one another, but only with the PAN coordinator (e.g., a unicast message transmission). Only four nodes were used because the open source implementation used became unstable with more than four nodes in the network. However, it is important to note that these attacks are *independent* of the number of nodes deployed in the network.

4.2 Hardware and Software Components

We used four Tmote Sky motes [29] based on TelosB platform: one for PAN coordinator, two for legitimate nodes, and one for the malicious node. In addition, we used the Texas Instruments (TI) CC2420 Evaluation Board/Evaluation Module (EB/EM) [30] in conjunction with the TI Chipcon packet sniffer [31] to capture and analyze packet traffic in the network. For the attack implementation, we used a 802.15.4 open source supporting a beacon-enabled mode from Open-ZB [32]. In particular, we used the open source v1.2 in conjunction with TinyOS v1.15 [33]. Figure 4.1 shows Tmote Sky motes and CC2420 EB/EM. Figure 4.2 shows examples of captured packets from the TI Chipcon packet sniffer.

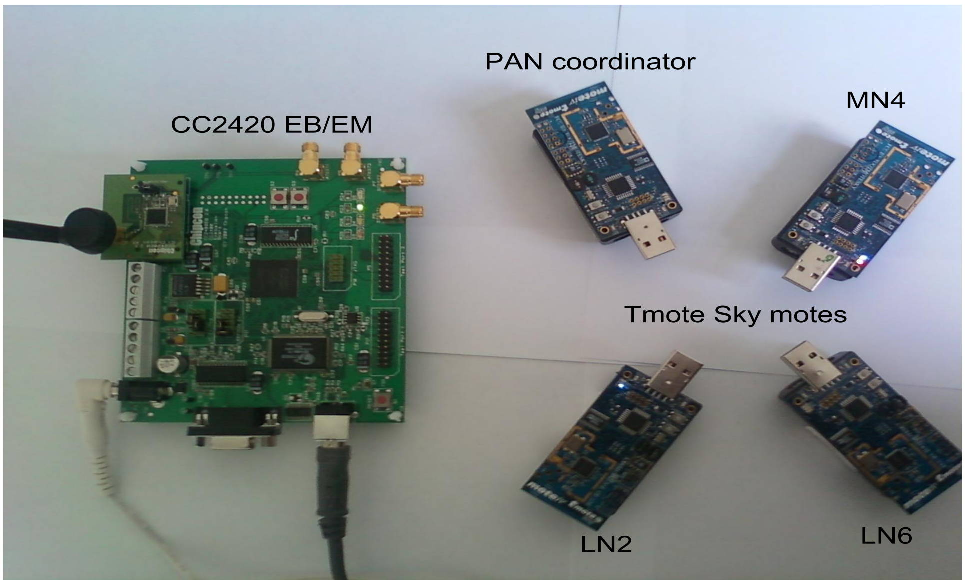
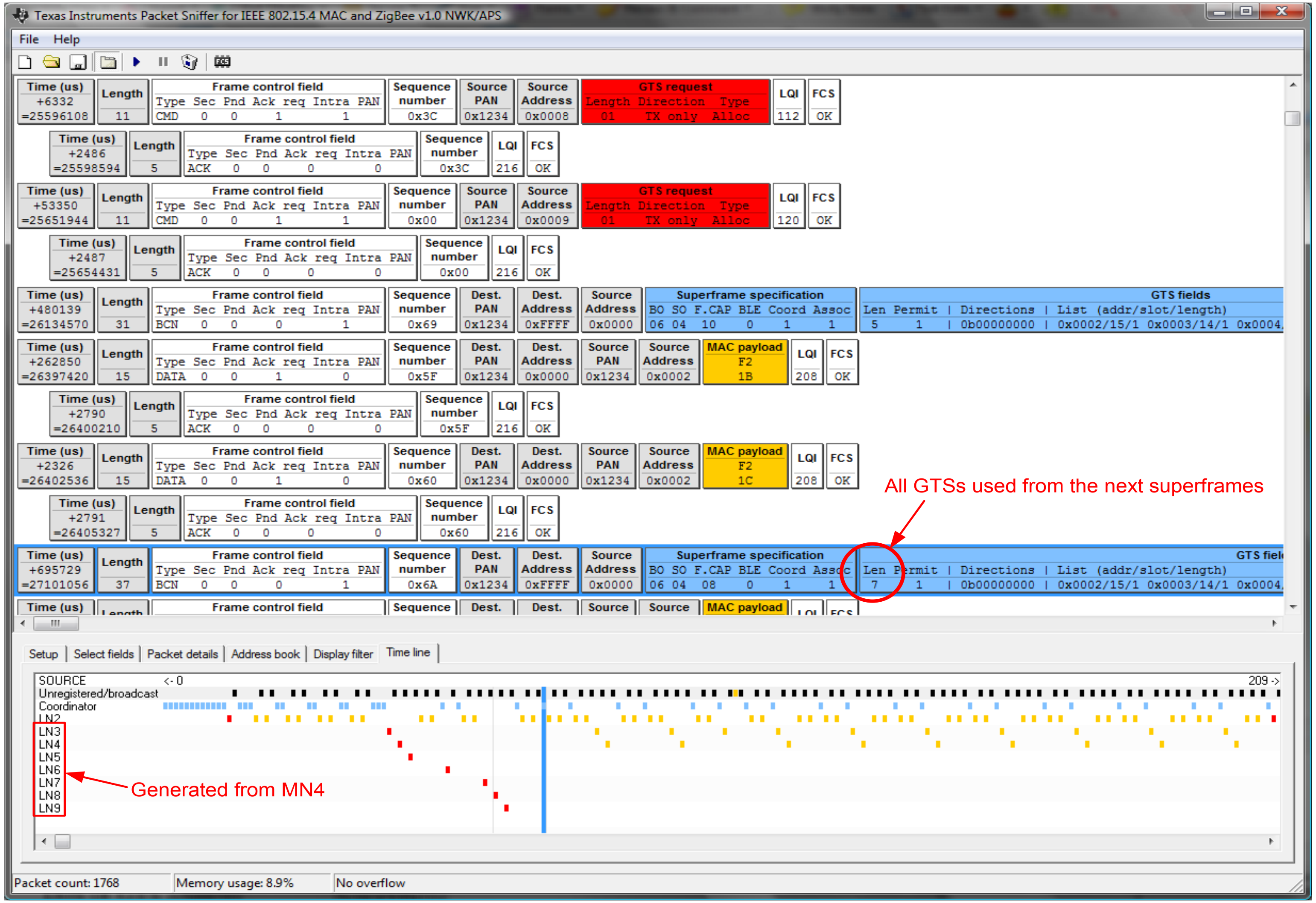


Figure 4.1. Tmote Sky motes and CC2420 EB/EM.



```
Time (us) | Length | Frame control field                  | Sequence | Source  | Source  | GTS request                   | LQI | FCS
+6332     |        | Type Sec Pnd Ack req Intra PAN       | number   | PAN     | Address | Length Direction  Type        |     |
=25596108 | 11     | CMD   0   0     1          1         | 0x3C     | 0x1234  | 0x0008  | 01       TX only  Alloc       | 112 | OK

    Time (us) | Length | Frame control field                  | Sequence | LQI | FCS
    +2486     |        | Type Sec Pnd Ack req Intra PAN       | number   |     |
    =25598594 | 5      | ACK   0   0     0          0         | 0x3C     | 216 | OK

Time (us) | Length | Frame control field                  | Sequence | Source  | Source  | GTS request                   | LQI | FCS
+53350    |        | Type Sec Pnd Ack req Intra PAN       | number   | PAN     | Address | Length Direction  Type        |     |
=25651944 | 11     | CMD   0   0     1          1         | 0x00     | 0x1234  | 0x0009  | 01       TX only  Alloc       | 120 | OK

    Time (us) | Length | Frame control field                  | Sequence | LQI | FCS
    +2487     |        | Type Sec Pnd Ack req Intra PAN       | number   |     |
    =25654431 | 5      | ACK   0   0     0          0         | 0x00     | 216 | OK

Time (us) | Length | Frame control field                  | Sequence | Dest.   | Dest.   | Source  | Superframe specification          | GTS fields
+480139   |        | Type Sec Pnd Ack req Intra PAN       | number   | PAN     | Address | Address | BO SO F.CAP BLE Coord Assoc       | Len Permit | Directions | List (addr/slot/length)
=26134570 | 31     | BCN   0   0     0          1         | 0x69     | 0x1234  | 0xFFFF  | 0x0000  | 06 04  10    0    1     1         |  5     1   | 0b00000000 | 0x0002/15/1 0x0003/14/1 0x0004,

Time (us) | Length | Frame control field                  | Sequence | Dest.   | Dest.   | Source  | Source  | MAC payload | LQI | FCS
+262850   |        | Type Sec Pnd Ack req Intra PAN       | number   | PAN     | Address | PAN     | Address | F2          |     |
=26397420 | 15     | DATA  0   0     1          0         | 0x5F     | 0x1234  | 0x0000  | 0x1234  | 0x0002  | 1B          | 208 | OK

    Time (us) | Length | Frame control field                  | Sequence | LQI | FCS
    +2790     |        | Type Sec Pnd Ack req Intra PAN       | number   |     |
    =26400210 | 5      | ACK   0   0     0          0         | 0x5F     | 216 | OK

Time (us) | Length | Frame control field                  | Sequence | Dest.   | Dest.   | Source  | Source  | MAC payload | LQI | FCS
+2326     |        | Type Sec Pnd Ack req Intra PAN       | number   | PAN     | Address | PAN     | Address | F2          |     |
=26402536 | 15     | DATA  0   0     1          0         | 0x60     | 0x1234  | 0x0000  | 0x1234  | 0x0002  | 1C          | 208 | OK

    Time (us) | Length | Frame control field                  | Sequence | LQI | FCS
    +2791     |        | Type Sec Pnd Ack req Intra PAN       | number   |     |
    =26405327 | 5      | ACK   0   0     0          0         | 0x60     | 216 | OK

Time (us) | Length | Frame control field                  | Sequence | Dest.   | Dest.   | Source  | Superframe specification          | GTS fiel
+695729   |        | Type Sec Pnd Ack req Intra PAN       | number   | PAN     | Address | Address | BO SO F.CAP BLE Coord Assoc       | Len Permit | Directions | List (addr/slot/length)
=27101056 | 37     | BCN   0   0     0          1         | 0x6A     | 0x1234  | 0xFFFF  | 0x0000  | 06 04  08    0    1     1         |  7     1   | 0b00000000 | 0x0002/15/1 0x0003/14/1 0x0004,

Time (us) | Length | Frame control field                  | Sequence | Dest.   | Dest.   | Source  | Source  | MAC payload | LQI | FCS
```

All GTSs used from the next superframes

Generated from MN4

Figure 4.2. Captured packets from TI Chipcon packet sniffer.

Chapter 5

OVERVIEW OF ATTACKS

In this chapter, we introduce the attack model and illustrate the overview of the attacks based on the model. We present a total of six attacks and categorize them according to the characteristics of the attacks. Table 5.1 lists the attacks and their characteristics.

5.1 Attack Model

Similar to the threat models defined in [26] and [34], we assume that a malicious node behaves badly as a mote-class, inside, and active attacker. We deploy one Tmote Sky mote as a malicious node that has the same capabilities as the legitimate nodes. The malicious node is located near legitimate nodes in the beacon-enabled 802.15.4 network. The malicious node listens to beacons from the PAN coordinator to get synchronization information of the network and GTS (de)allocation requests from legitimate nodes in the passive phase. In the active phase, since an authentication between legitimate nodes and the PAN coordinator may not be present due to higher communication cost, it is easier for the malicious node to impersonate either legitimate nodes or the PAN coordinator and to attack the vulnerabilities of the beacon broadcasts and the GTS management scheme. Table 5.1 presents a summary of the attacks and the vulnerabilities of the beacon-enabled 802.15.4 MAC.

Table 5.1. Attacks and Their Characteristics. LN is legitimate nodes. PC is the PAN coordinator.

Attacks		Unauthenticated Message (Direction)	Node impersonated	Vulnerabilities	Message transmission types
Synchronization attack		Beacon (LN ← PC)	Existing PAN coordinator	Beacon broadcast	Broadcast
DoS of data transmission	Impersonating a legitimate node	GTS allocation request (LN → PC)	Existing node IDs	GTS management scheme	Unicast
	Impersonating the PAN coordinator	Beacon (LN ← PC)	Existing PAN coordinator		
False data injection		Data (LN → PC)	Existing node IDs		
DoS of GTS requests		GTS allocation request (LN → PC)	Non-existing node IDs		
Stealing network bandwidth					

5.2 Impersonating Existing Identities in the PAN

In this category, we describe four attacks. The first attack presented is the synchronization attack. In this attacks, legitimate nodes are lead to synchronize their SF timeslots with the manipulated beacons from the malicious node. The next two attacks block data transmission from legitimate nodes in the PAN that want to gain GTSs and transmit time-sensitive data in the slots. The fourth attack injects false sensed data into the traffic stream from the legitimate node to the PAN coordinator during the CFP.

5.2.1 Synchronization Attack

This attack influences all the nodes in the network concurrently, whereas the other attacks can affect only one or a few legitimate nodes. The malicious node first impersonates the PAN coordinator's ID and uses the same PAN ID as that of the PAN coordinator. The malicious node manipulates two important parameters: BO and SO as shown in Figure 3.1. To compete with the beacons from the PAN coordinator, the malicious node sets the BO less than or equal to that of the PAN coordinator and the SO less than that of the PAN coordinator. Figure 5.1 shows two different SF sequences. Figure 5.1 (b) has short SF intervals, as compared to that of the legitimate coordinator (shown in Figure 5.1 (a)), due to the smaller BO and SO. Thus, when the legitimate nodes process the beacons from both the PAN coordinator and the malicious node, they synchronize their SF timeslots with the manipulated BO and SO (e.g., 4 and 2 respectively) if the beacons from the malicious node arrives immediately after those from the PAN coordinator.

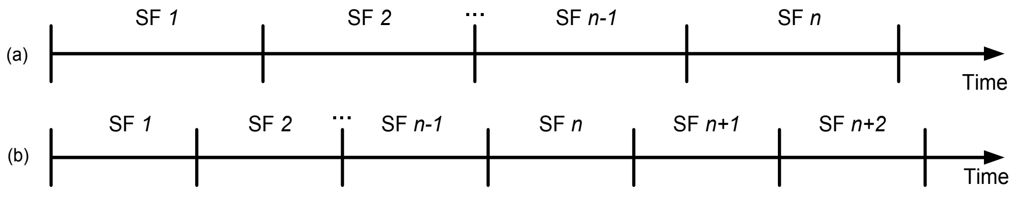


Figure 5.1. The SFs, (a) and (b), configured by the PAN coordinator and the malicious node respectively. For example, the BO and SO of (a) are set to 6 and 4 respectively, and the BO and SO of (b) are set to 4 and 2 respectively.

5.2.2 DoS of Data Transmission

Impersonating a legitimate node: If a malicious node is in the transmission range of the PAN coordinator, it is easily able to obtain the IDs of active legitimate nodes in the PAN. The malicious node also knows whether or not a legitimate node tries to transmit its sensed data during the CFP by looking at the GTS allocation requests or the beacons. In this attack, the malicious node impersonates the active legitimate nodes in the PAN and sends GTS deallocation requests using the legitimate nodes' IDs to the PAN coordinator. Figure 5.2 (a) shows an example of this attack. While two legitimate nodes request GTS allocation to transmit data in the next SF's CFP, the malicious node can terminate the data transmissions of the legitimate nodes by sending a GTS deallocation request with the legitimate nodes' IDs. Since the PAN coordinator receives the GTS deallocation request while processing the GTS allocation from the legitimate nodes, it ignores the GTS allocation coming first and does not assign any GTS to the legitimate nodes. As a result, the legitimate nodes that do not have any assigned GTS cannot transmit its sensed data.

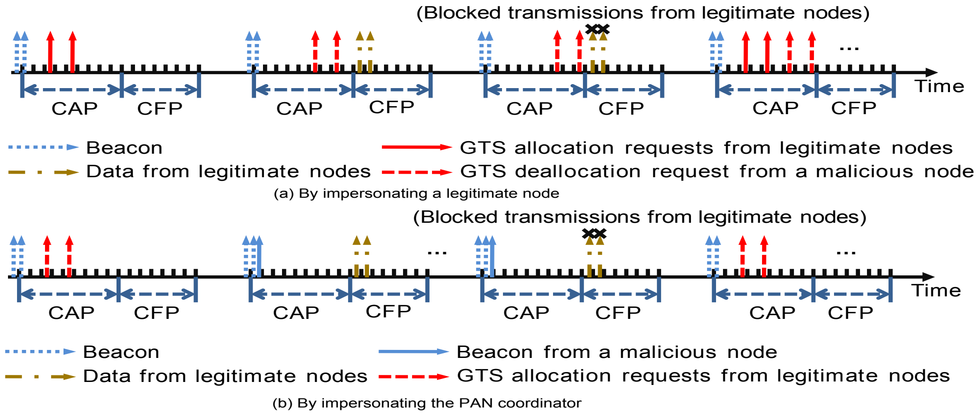

(Blocked transmissions from legitimate nodes)
...
Time
CAP
CFP
Beacon
Data from legitimate nodes
GTS allocation requests from legitimate nodes
GTS deallocation request from a malicious node
(a) By impersonating a legitimate node
(Blocked transmissions from legitimate nodes)
...
Time
CAP
CFP
Beacon
Data from legitimate nodes
Beacon from a malicious node
GTS allocation requests from legitimate nodes
(b) By impersonating the PAN coordinator

Figure 5.2. A malicious node blocking a legitimate node sending data during CFP. (a) represents that the malicious node pretends to be a legitimate node. (b) represents that the malicious node pretends to be the PAN coordinator.

Impersonating the PAN coordinator: The previous attack impersonates the legitimate nodes' IDs to cause the PAN coordinator to deallocate the GTSs. In this attack, the malicious node impersonates the PAN coordinator and broadcasts manipulated beacons that do not include any GTS descriptor. When the legitimate nodes request GTS allocation to transmit its sensed data, they wait for beacons coming with the GTS descriptors that tell them the assigned GTS information (e.g., address, slot, and length as shown in Figure 3.2 (c)). If the beacons that the legitimate nodes receive do not have any GTS descriptor, the nodes assume that they cannot transmit the sensed data to the PAN coordinator due to no GTS being allocated. Thus, the malicious node impersonating the PAN coordinator keeps sending a manipulated beacon without GTS descriptors right after the PAN coordinator broadcasts a beacon with GTS descriptors. Since the legitimate nodes just process the beacon coming last if there are more than one beacon received within the proper boundary of the timeslot, it is told that no GTS is assigned by the manipulated beacons from the malicious node coming last and is not transmitting data to the PAN coordinator as shown in Figure 5.2 (b).

5.2.3 False Data Injection

In this attack, the malicious node identifies which legitimate node has not requested GTS allocation by looking at the GTS descriptors of beacons. Then, the malicious node chooses the legitimate node's ID that does not have any GTS allocation request and sends a GTS allocation request using that ID. After it confirms that a GTS is allocated by the PAN coordinator, the malicious node sends false data with the ID to the PAN coordinator during the CFP while the legitimate node sends its sensed data during the CAP. After checking the node's ID, the PAN coordinator regards the false data as time-sensitive ones from the node due to being sent during the CFP. Then, it can update with the false data sent by the malicious node. Figure 5.3 shows how this attack works; for instance, when a legitimate node is transmitting current temperature data during the CAP, the malicious node sends a GTS allocation request with the spoofed ID, pretends to be another legitimate node, and can inject false temperature data during the CFP.

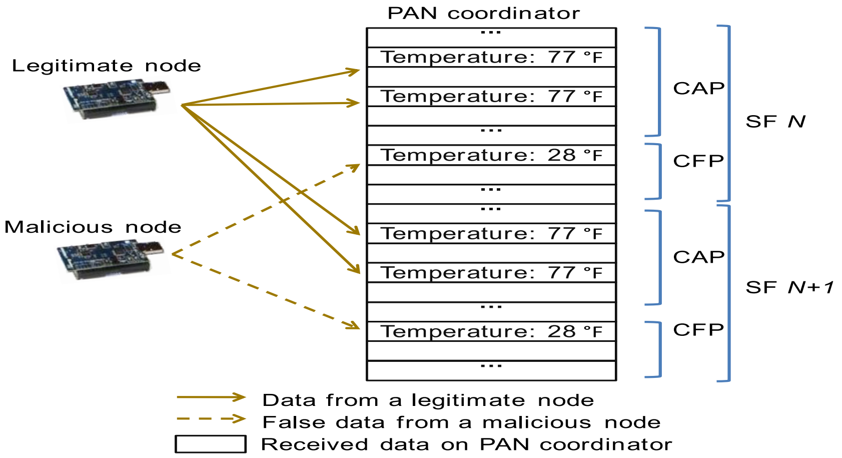


Figure 5.3. A malicious node sending false temperature to the PAN coordinator.

5.3 Impersonating Non-existing Identities in the PAN

In this category, a malicious node forges up to 7 different IDs depending on the maximum number of available GTSs. The two attacks presented in this section perform exhaustion and unfairness attacks by occupying all 7 GTSs and not allowing legitimate nodes to reserve GTSs.

5.3.1 DoS of GTS Requests

To perform this attack, a malicious node continuously monitors the available GTS slots with the intent of completely occupying them. Then, the attacker sends several GTS allocation requests to fill up all the available GTSs in the SF. The advantage of this attack is that the malicious node can reduce its energy consumption, because once it occupies all 7 GTSs, it does not need to send out any data or commands. The malicious node simply dissects beacon frames to see if the PAN coordinator performs the preventative action for the CAP maintenance. Figure 5.4 shows that after legitimate nodes A and B send GTS deallocation requests, the malicious node completely fills all 7 GTSs with two additional GTS allocation requests. The goal of this attack is *not* for the attacker to use the bandwidth requested, rather it is to prevent the legitimate nodes from transmitting data during the CFP.

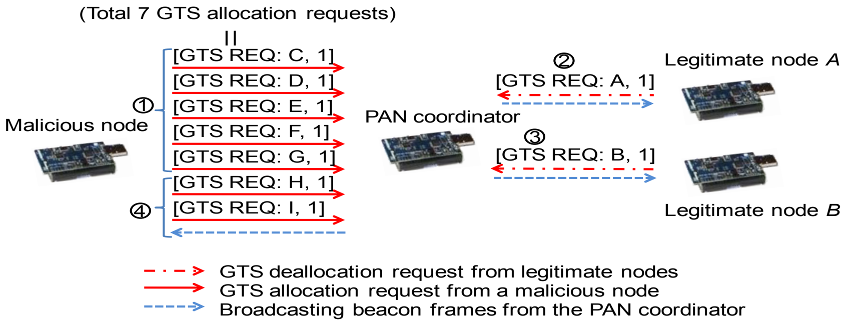


Figure 5.4. A malicious node filling up all 7 GTSs. 1: the malicious node sends five GTS allocation requests. 2 and 3: legitimate node A and B send GTS deallocation requests. 4: the malicious node sends the rest of GTS allocation requests.

5.3.2 Stealing Network Bandwidth

Similar to the DoS of GTS requests, in this attack, an attacker observes the GTS list in order to eventually occupy the available GTS slots. However, in this attack, the malicious node sends data at the assigned timeslots. The purpose of data transmission is to prevent the PAN coordinator from dropping the assigned GTSs. As shown in Figure 5.5, the second CFP has data transmitted from both legitimate nodes and a malicious node. However, since legitimate nodes send GTS deallocation requests during the second CAP, the malicious node sends a GTS allocation request to occupy the new free GTS. Eventually, only the malicious node sends data during the fourth CFP. The timeslots will never be vacant during the CFP of every SF, which can cause both exhaustion and unfairness against legitimate nodes. This also affects the PAN coordinator who cannot go into sleep mode (denial of sleep attack [35]) due to the malicious node continuously sending data.

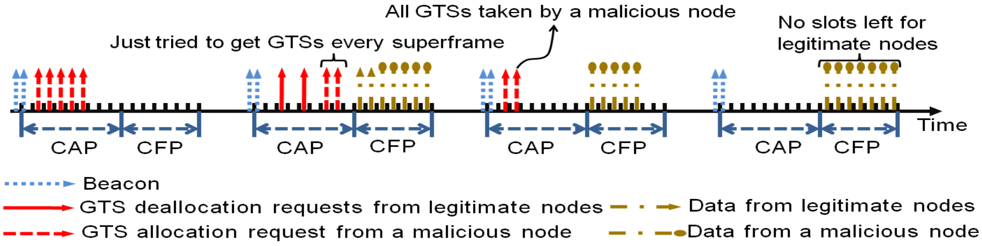

All GTSs taken by a malicious node
Just tried to get GTSs every superframe
No slots left for legitimate nodes
CAP
CFP
CAP
CFP
CAP
CFP
CAP
CFP
Time
Beacon
GTS deallocation requests from legitimate nodes
GTS allocation request from a malicious node
Data from legitimate nodes
Data from a malicious node

Figure 5.5. A malicious node stealing all 7 GTSs during CFP.

Chapter 6

IMPLEMENTATION OF ATTACKS

In this chapter, we introduce the application layer and MAC layer modules that were implemented to execute the six attacks described in the previous chapter. We explain in detail how the malicious node runs the attacks in the PAN.

6.1 Attack Modules for Implementation

We have implemented our attacks based on the existing modules provided by Open-ZB. Given the modules in the MAC layer of the 802.15.4 protocol, we mainly modified the source code of MAC layer for our attacks and added a malicious application (MAC misbehavior app) as shown in Figure 6.1. The modified MAC layer and the malicious application target the vulnerabilities of the GTS management scheme described in Chapter 3.

We have two options for implementing the MAC misbehavior attacks: the first option is to implement a module in the application layer, while the second option relies on the implementation of modules in the application and MAC layers. In the application layer implementation, most operations of the attacks are controlled and executed in *MAC misbehavior app*. For instance, *MAC misbehavior app* calls *MLME_SET()* to set existing IDs or multiple non-existing faked IDs in the malicious node and calls *MLME_GTS_request()* in *Mal-GTS management* to send GTS (de)allocation requests with manipulated IDs as illustrated in Figure 6.2. However, since *MAC misbehavior app* cannot determine the appropriate time to send GTS (de)allocation requests or manipulated beacons without getting the information from the MAC layer, it has to either send repeated GTS requests and beacons frequently or

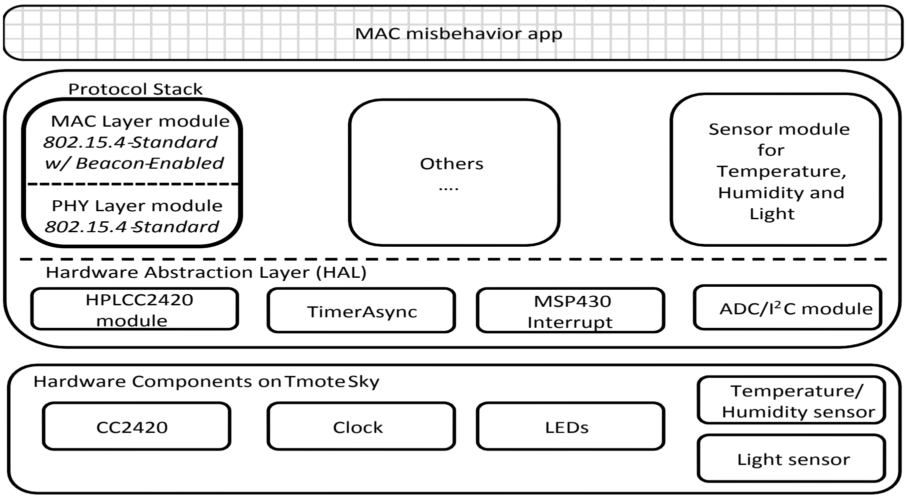


Figure 6.1. The software and hardware modules of the Tmote Sky mote. The software modules consist of TinyOS v1.15 and the protocol stack of Open-ZB v1.2. The shaded region represents the modified modules for the malicious node.

get triggered by the MAC layer to send them. Either way has high performance overhead that can consume the battery of Tmote Sky mote due to high transmission frequency or increased function calls between the application and MAC layers.

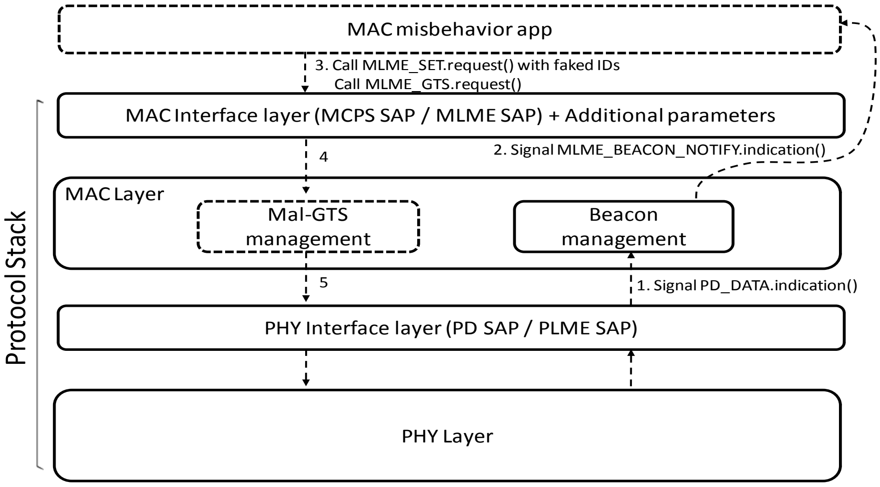

MAC misbehavior app
3. Call MLME_SET.request() with faked IDs
Call MLME_GTS.request()
MAC Interface layer (MCPS SAP / MLME SAP) + Additional parameters
4
2. Signal MLME_BEACON_NOTIFY.indication()
MAC Layer
Mal-GTS management
Beacon management
5
1. Signal PD_DATA.indication()
PHY Interface layer (PD SAP / PLME SAP)
PHY Layer
Protocol Stack

Figure 6.2. The attacks implemented and controlled in the application layer.

For this reason, we implement the MAC misbehavior attacks by adding *Mal-PD_DATA management* in the MAC layer as shown in Figure 6.3. *Mal-PD_DATA management* intercepts a function, *PD_DATA.indication()*, which indicates all packets (e.g., beacon, command (GTS request), data) of the communication in the PAN. Then, it directly executes the attacks in the MAC layer. For instance, if the packet is a beacon from the PAN coordinator, *Mal-PD_DATA management* looks at available GTSs and directly calls *Mal-GTS management* to fill all remaining GTSs. If the packet is a GTS allocation request from a legitimate node, *Mal-PD_DATA management* informs *Mal-GTS management* of the legitimate node's ID. Then, *Mal-GTS management* sends a GTS deallocation request with the ID. This allows our attacks to be more efficiently executed with lower communication overhead.

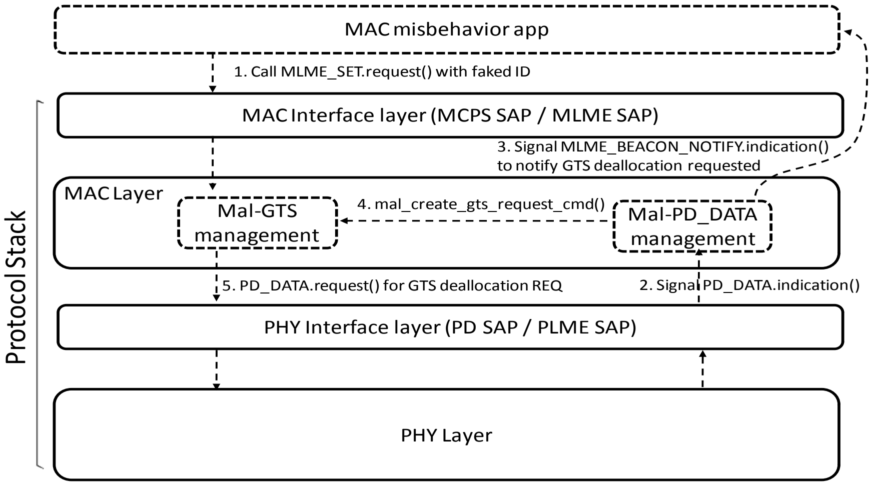

MAC misbehavior app
1. Call MLME_SET.request() with faked ID
MAC Interface layer (MCPS SAP / MLME SAP)
3. Signal MLME_BEACON_NOTIFY.indication()
to notify GTS deallocation requested
MAC Layer
Mal-GTS management
4. mal_create_gts_request_cmd()
Mal-PD_DATA management
5. PD_DATA.request() for GTS deallocation REQ
2. Signal PD_DATA.indication()
PHY Interface layer (PD SAP / PLME SAP)
PHY Layer
Protocol Stack

Figure 6.3. The attacks improved by implementing and controlling in the MAC layer with the *Mal-PD_DATA* management module.

6.2 Impersonating Existing Identities in the PAN

In this section, we assume that there is one PAN coordinator, two legitimate nodes (LN2 and LN6), and one malicious node (MN4) as shown in Figures 6.4-6.7. MN4 impersonates the IDs of LN2, LN6, and the PAN coordinator by eavesdropping on the traffic in the PAN.

6.2.1 Synchronization Attack

In this attack, MN4 recognizes that the BO is 6 and the SO is 4. Next, it broadcasts beacons with lower or equal value than the legitimate BO and SO (according to the notation, $0 \leq \text{SO} \leq \text{BO} \leq 14$, in the 802.15.4 standard) until the manipulated beacons lead the node to synchronize its SF timeslots using the BO and SO (e.g., 4 and 2 respectively) from MN4. To prevent LN2 from dropping the manipulated beacons, MN4 should be careful to send the first beacon that could be processed within the boundary of the timeslot when LN2 processes the beacon from the PAN coordinator. Thus, in the 3rd SF in Figure 6.4, the manipulated beacon with BO = 4 and SO = 2 is sent right after MN4 receives the legitimate beacon from the PAN coordinator. LN2 receives the legitimate beacon first and the manipulated beacon last before finishing the beacon process. Then, LN2 actually synchronizes its SF timeslots with BO = 4 and SO = 2 in the 3rd SF. From the 3rd SF, LN2 can transmit its sensed data to MN4 during the CFP that is provided by the MN4's SFs.

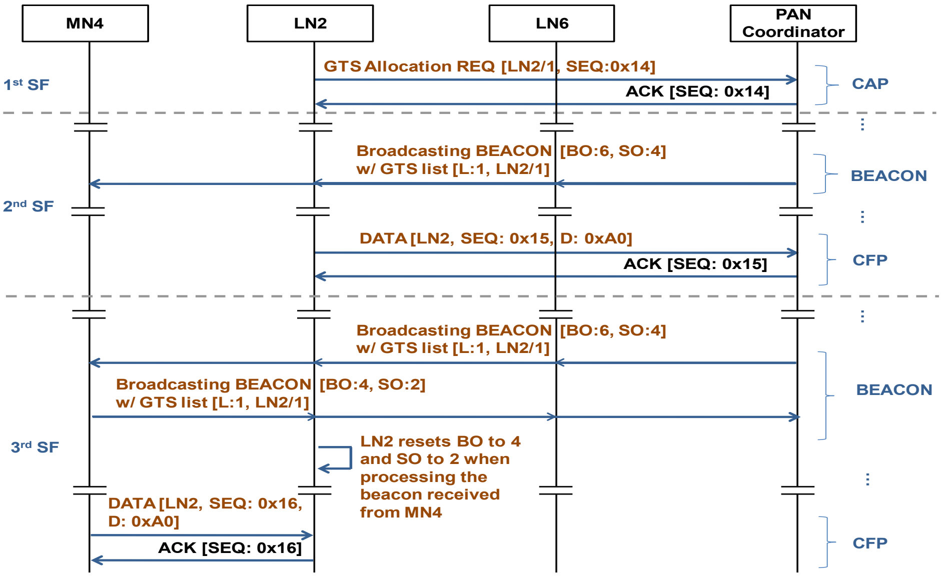

MN4
LN2
LN6
PAN Coordinator
1st SF
GTS Allocation REQ [LN2/1, SEQ:0x14]
ACK [SEQ: 0x14]
CAP
Broadcasting BEACON [BO:6, SO:4]
w/ GTS list [L:1, LN2/1]
BEACON
2nd SF
DATA [LN2, SEQ: 0x15, D: 0xA0]
ACK [SEQ: 0x15]
CFP
Broadcasting BEACON [BO:6, SO:4]
w/ GTS list [L:1, LN2/1]
Broadcasting BEACON [BO:4, SO:2]
w/ GTS list [L:1, LN2/1]
BEACON
3rd SF
LN2 resets BO to 4 and SO to 2 when processing the beacon received from MN4
DATA [LN2, SEQ: 0x16, D: 0xA0]
ACK [SEQ: 0x16]
CFP

Figure 6.4. The sequence for synchronization attack.

6.2.2 DoS of Data Transmission

Impersonating a legitimate node: As shown in Figure 6.5, this attack works through two SFs. In the first SF, LN2 and LN6 send GTS allocation requests to the PAN coordinator to reserve one GTS. Then, MN4 immediately sends GTS deallocation requests with the impersonated LN2 and LN6's IDs in the same CAP right after their GTS allocation requests. The PAN coordinator removes LN2 and LN6 from the GTS list and does not receive data during the CFP of the next SF. Since LN2 and LN6 are not allocated to GTSs, they are not able to send their messages during the CFP.

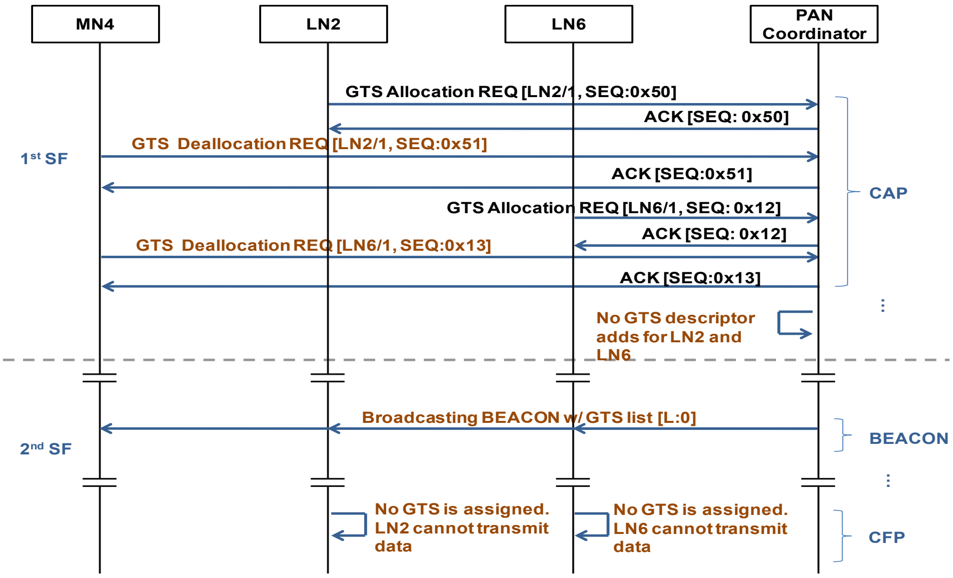

MN4
LN2
LN6
PAN Coordinator
GTS Allocation REQ [LN2/1, SEQ:0x50]
ACK [SEQ: 0x50]
GTS Deallocation REQ [LN2/1, SEQ:0x51]
1st SF
ACK [SEQ:0x51]
GTS Allocation REQ [LN6/1, SEQ: 0x12]
ACK [SEQ: 0x12]
GTS Deallocation REQ [LN6/1, SEQ:0x13]
ACK [SEQ:0x13]
CAP
No GTS descriptor adds for LN2 and LN6
Broadcasting BEACON w/ GTS list [L:0]
BEACON
2nd SF
No GTS is assigned. LN2 cannot transmit data
No GTS is assigned. LN6 cannot transmit data
CFP

Figure 6.5. The sequence for DoS of data transmission by impersonating a legitimate node.

Impersonating the PAN coordinator: In this attack, MN4 uses the same values of the BO and SO as the PAN coordinator (obtained by eavesdropping on the normal beacons in the PAN). However, while LN2 requests GTS allocation and transmits its data at an assigned GTS, MN4 broadcasts the manipulated beacons without GTS descriptors right after receiving the beacons from the PAN coordinator in the 3rd SF as shown in Figure 6.6. LN2 processes the manipulated beacon that is received last. Since the beacon does not include GTS descriptors, LN2 is not able to transmit its sensed data due to no available GTS assigned.

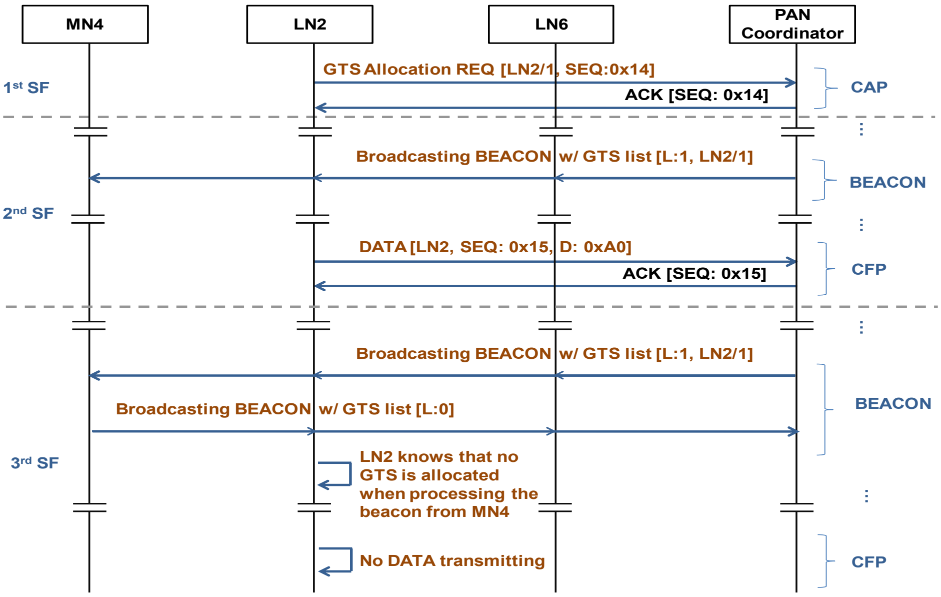

MN4
LN2
LN6
PAN Coordinator
1st SF
GTS Allocation REQ [LN2/1, SEQ:0x14]
ACK [SEQ: 0x14]
CAP
Broadcasting BEACON w/ GTS list [L:1, LN2/1]
BEACON
2nd SF
DATA [LN2, SEQ: 0x15, D: 0xA0]
ACK [SEQ: 0x15]
CFP
Broadcasting BEACON w/ GTS list [L:1, LN2/1]
Broadcasting BEACON w/ GTS list [L:0]
BEACON
3rd SF
LN2 knows that no GTS is allocated when processing the beacon from MN4
No DATA transmitting
CFP

Figure 6.6. The sequence for DoS of data transmission by impersonating the PAN coordinator.

6.2.3 False Data Injection

Unlike DoS of data transmission, this attack exploits GTS allocation requests to transmit false data. Figure 6.7 shows such a case where LN2 has already been assigned to one GTS. In this case, MN4 starts after LN2 sends a GTS deallocation request in the first SF. Then, the PAN coordinator removes LN2's ID on the GTS list of the next beacon. Since MN4 is aware that LN2 is not in the GTS list, it immediately tries to get one GTS by sending a GTS allocation request with LN2's ID. Once MN4 successfully takes the GTS, it starts sending false data with LN2's ID in the third SF.

6.3 Impersonating Non-existing Identities in the PAN

For forging non-existing IDs, we also have one PAN coordinator, two legitimate nodes (LN2 and LN6), and one malicious node (MN4) that generates false IDs that are different from LN2 and LN6. In this case, MN4 eavesdrops on the beacons to learn what IDs do not belong in the PAN.

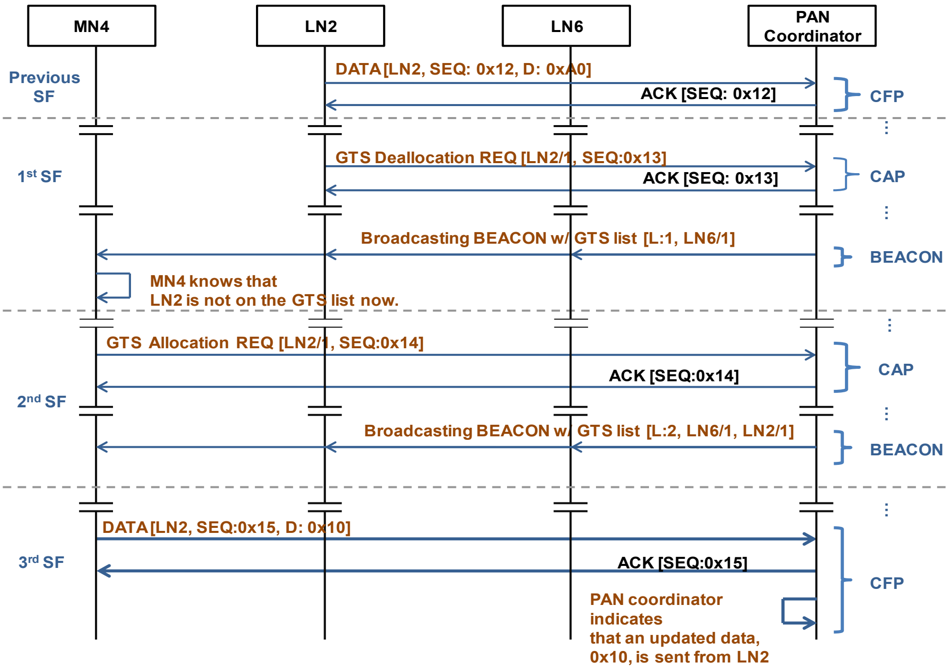

MN4
LN2
LN6
PAN Coordinator
Previous SF
DATA [LN2, SEQ: 0x12, D: 0xA0]
ACK [SEQ: 0x12]
CFP
1st SF
GTS Deallocation REQ [LN2/1, SEQ:0x13]
ACK [SEQ: 0x13]
CAP
Broadcasting BEACON w/ GTS list [L:1, LN6/1]
BEACON
MN4 knows that
LN2 is not on the GTS list now.
GTS Allocation REQ [LN2/1, SEQ:0x14]
ACK [SEQ:0x14]
CAP
2nd SF
Broadcasting BEACON w/ GTS list [L:2, LN6/1, LN2/1]
BEACON
DATA [LN2, SEQ:0x15, D: 0x10]
3rd SF
ACK [SEQ:0x15]
CFP
PAN coordinator
indicates
that an updated data,
0x10, is sent from LN2

Figure 6.7. The sequence for false data injection.

6.3.1 DoS of GTS Requests

As shown in Figure 6.8, this attack needs several SFs to allow MN4 to fill all 7 GTSs. In each SF, MN4 knows how many GTSs are available and sends GTS allocation requests in order to reserve the remaining slots of GTSs. Once MN4 takes all 7 GTSs, it stops sending GTS allocation requests to reduce its energy consumption. It then monitors the beacons to see if the PAN coordinator drops the unused GTSs by a preventative action for the CAP maintenance. If this occurs, MN4 will start sending GTS allocation requests again.

6.3.2 Stealing Network Bandwidth

Figure 6.9 shows that a malicious node takes the last slot out of 7 GTSs, 6 slots of which were already assigned to the malicious node. Then, it can consume all 7 GTSs during the CFP to transmit data. The difference from the previous DoS of GTS Requests is that since this attack continues to transmit data at each timeslot of the CFP, the PAN coordinator will not take any preventative action for the CAP maintenance.

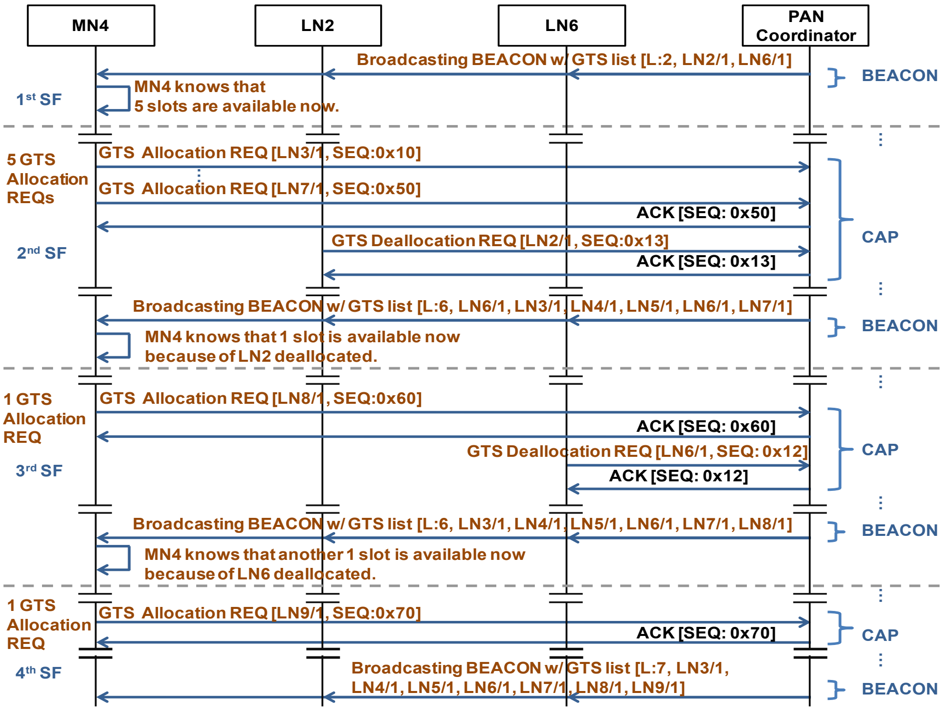

MN4
LN2
LN6
PAN Coordinator
Broadcasting BEACON w/ GTS list [L:2, LN2/1, LN6/1]
BEACON
1st SF
MN4 knows that
5 slots are available now.
5 GTS Allocation REQs
GTS Allocation REQ [LN3/1, SEQ:0x10]
GTS Allocation REQ [LN7/1, SEQ:0x50]
ACK [SEQ: 0x50]
2nd SF
GTS Deallocation REQ [LN2/1, SEQ:0x13]
ACK [SEQ: 0x13]
CAP
Broadcasting BEACON w/ GTS list [L:6, LN6/1, LN3/1, LN4/1, LN5/1, LN6/1, LN7/1]
BEACON
MN4 knows that 1 slot is available now
because of LN2 deallocated.
1 GTS Allocation REQ
GTS Allocation REQ [LN8/1, SEQ:0x60]
ACK [SEQ: 0x60]
3rd SF
GTS Deallocation REQ [LN6/1, SEQ: 0x12]
ACK [SEQ: 0x12]
CAP
Broadcasting BEACON w/ GTS list [L:6, LN3/1, LN4/1, LN5/1, LN6/1, LN7/1, LN8/1]
BEACON
MN4 knows that another 1 slot is available now
because of LN6 deallocated.
1 GTS Allocation REQ
GTS Allocation REQ [LN9/1, SEQ:0x70]
ACK [SEQ: 0x70]
CAP
4th SF
Broadcasting BEACON w/ GTS list [L:7, LN3/1,
LN4/1, LN5/1, LN6/1, LN7/1, LN8/1, LN9/1]
BEACON

Figure 6.8. The sequence for DoS of GTS requests.

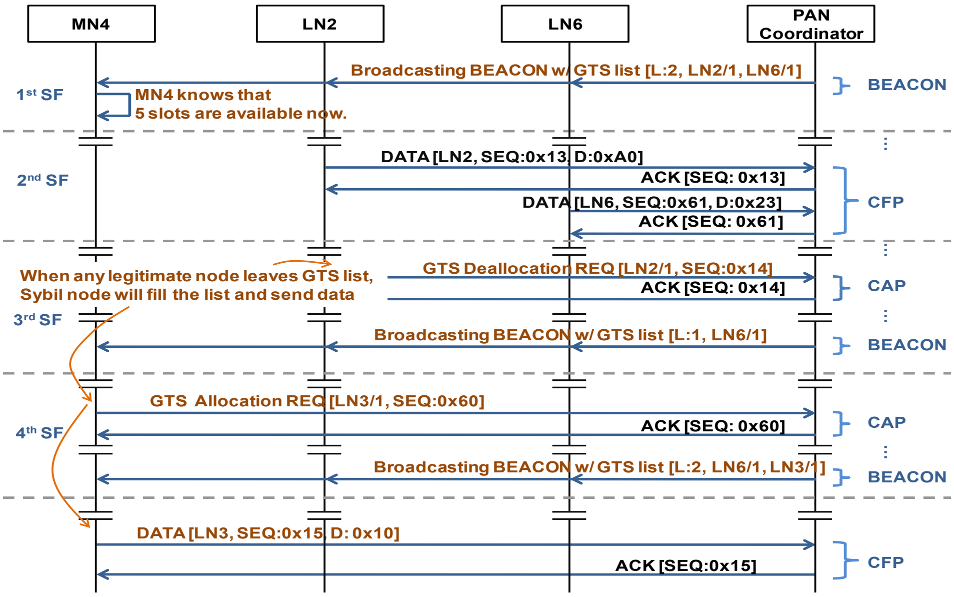

MN4
LN2
LN6
PAN Coordinator
Broadcasting BEACON w/ GTS list [L:2, LN2/1, LN6/1]
BEACON
1st SF
MN4 knows that
5 slots are available now.
DATA [LN2, SEQ:0x13, D:0xA0]
ACK [SEQ: 0x13]
2nd SF
DATA [LN6, SEQ:0x61, D:0x23]
ACK [SEQ: 0x61]
CFP
When any legitimate node leaves GTS list,
Sybil node will fill the list and send data
GTS Deallocation REQ [LN2/1, SEQ:0x14]
ACK [SEQ: 0x14]
CAP
3rd SF
Broadcasting BEACON w/ GTS list [L:1, LN6/1]
BEACON
GTS Allocation REQ [LN3/1, SEQ:0x60]
ACK [SEQ: 0x60]
4th SF
CAP
Broadcasting BEACON w/ GTS list [L:2, LN6/1, LN3/1]
BEACON
DATA [LN3, SEQ:0x15, D: 0x10]
ACK [SEQ:0x15]
CFP

Figure 6.9. The sequence for stealing network bandwidth.

Chapter 7

ATTACK ANALYSIS

We have verified our implementation with the TI packet sniffer [31] to monitor the packet transmission while each attack is running. We employ the PAN coordinator to log humidity and temperature data sent by a legitimate node during both the CAP and the CFP. The throughput given in Figures 7.1, 7.2, and 7.4 are based on the total number of data in bytes divided by the elapsed time. The total data is counted only during the CFP. For each of the six attacks, we measured the packet transmission for 80 to 400 seconds depending on the complexity of each attack.

7.1 Synchronization attack

Figure 7.1 shows that LN2 synchronizes its SFs with the legitimate PAN coordinator broadcasting beacons (BO = 6, SO = 3) at first. However, it resynchronizes the SFs with MN4's beacons (BO = 4, SO = 2) around the 24-second mark. Figure 7.1 (a) illustrates that while the data throughput of LN2 is increasing up to 15 bps, MN4 starts broadcasting malicious beacons with BO (4) and SO (2) whereas the PAN coordinator broadcasts beacons with BO (6) and SO (3). At this moment, the MN4's data throughput starts increasing whereas the throughput to the PAN coordinator begins to decrease. The reason for lower data throughput from the 24-second mark is that the additional beacons from the PAN coordinator are still being sent within the shorter SFs of MN4, and LN2 transmits only 2 bytes in the SFs whereas it was previously sending more than 4 bytes in the SFs of the PAN coordinator. We can differentiate where the data is being transmitted from by checking the intervals between the first beacons and the first data packets in SFs. We measure the normal intervals in two different SFs. In one SF set by BO (6) and SO (3), the interval between the first beacon and the data packet is about 250 ms as shown in Figure 7.1 (b). In the other SF set by BO (4) and SO (2) of beacons, the interval is about 69 ms. Figure 7.1 (b) shows that the interval in the synchronization attack is reduced from about 250 ms to about 69 ms around the 24-second mark, which is the same time mark that MN4 begins to broadcast its fake beacons. As a result, LN2 perfectly synchronized the beacons with MN4 and transmitted data to MN4.

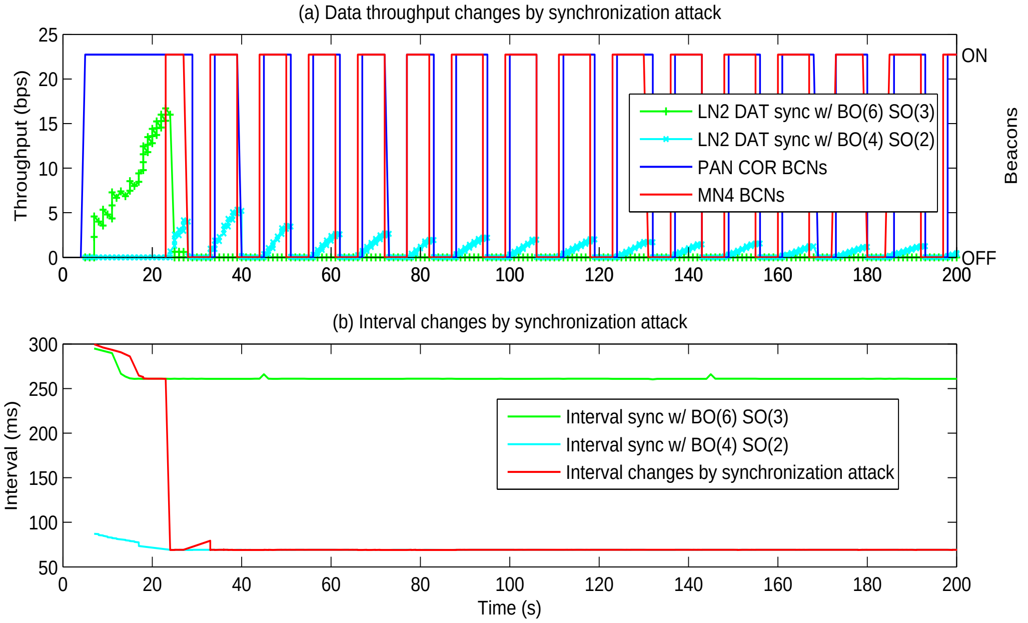


Figure 7.1. Legitimate node (LN2) data throughput changes after synchronizing with beacons from MN4. The intervals are between the first beacon and the first data packet in SFs.

7.2 DoS of Data Transmission

Impersonating a legitimate node: Figure 7.2 (a) shows the decline of data throughput on LN2 and LN6 while MN4 is sending GTS deallocation requests with LN2 and LN6's IDs. Around the 50-second mark of the experiment, MN4 sends two GTS deallocation requests back to back. It also sends the same two GTS deallocation requests whenever it receives a beacon-notification. Therefore, the data throughputs from LN2 and LN6 during the CFP drops to 0bps. Immediately after the 50-second mark, even though LN2 and LN6 try to send GTS allocation requests, the requests cannot be processed due to MN4 continuously sending GTS deallocation requests. By modifying the MAC layer (as discussed in Section 6.1), MN4 only sends GTS deallocation requests right after LN2 requests GTS allocation (around the 52-second mark and the 85-second mark in Figure 7.2 (b)). This reduces the transmission frequency of MN4 substantially. However, since the PAN coordinator only processes the last GTS deallocation request, this leads to the same result of blocking data transmission (no data transmission from the 50-second mark in Figure 7.2 (a) and (b)).

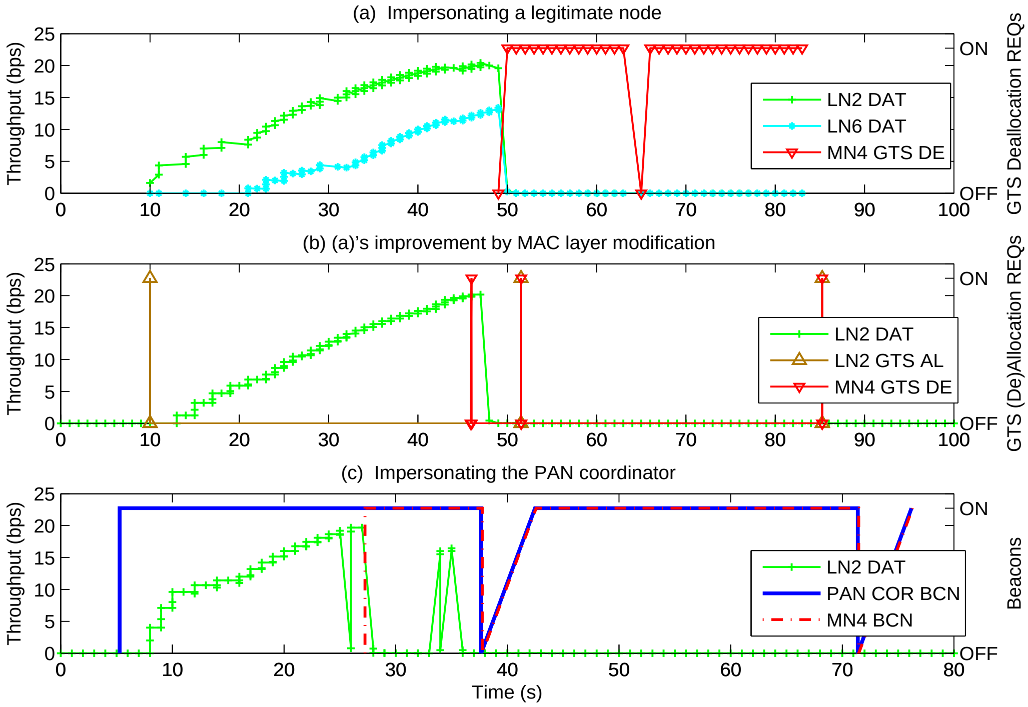


Figure 7.2. Legitimate nodes (LN2 and LN6) data throughput during CFP by a malicious node (MN4). LN2 DAT and LN6 DAT: Data from LN2 and LN6 and MN4 GTS: GTS deallocation requests from MN4.

Impersonating the PAN coordinator: Figure 7.2 (c) shows that MN4 starts sending the same beacons without GTS descriptors (i.e., no GTS assigned) around the 27-second mark, which immediately cripples LN2's throughput. Even though LN2 takes GTS information with the PAN coordinator in few SFs from 34-second to 36-second and transmits some data, it processes only the manipulated beacons from MN4 after around the 37-second mark. Thus, LN2 does not transmit its sensed data (no data transmission from the 37-second mark) because it assumes that no GTS is available due to the manipulated beacons without GTS descriptors. By impersonating the PAN coordinator, we produce the same blocking of data transmission as that shown in Figure 7.2 (a), (b), and (c).

7.3 False Data Injection

Figure 7.3 shows the change of humidity and temperature from LN2. We tested this attack inside a building, where the humidity and temperature conditions were approximately 41% and 72°F respectively. However, since MN4 impersonating LN2 sends false data readings of 90% for the humidity and 28°F for the temperature during the CFP, this results in fluctuations of the sensed data reported for 20 seconds around the 73 to 93-second mark. Since 28°F is below the freezing point, the false data of temperature might lead to a warning sign in a practical situation.

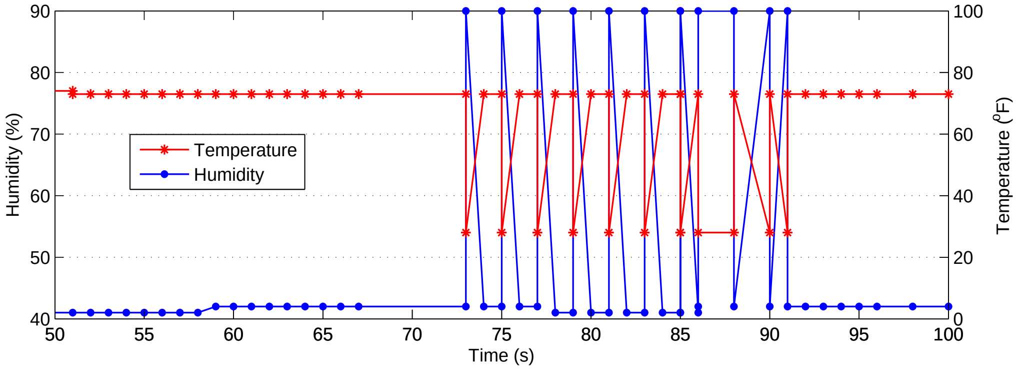


Figure 7.3. Fluctuation of humidity and temperature.

7.4 DoS of GTS Requests

Figure 7.4 (a) shows two instances of the DoS GTS Request attack. LN2 and MN4 start at the same time (around the 20-second mark). By sending a GTS request, LN2 quickly occupies one GTS and transmits data during the CFP. Similarly, MN4 quickly occupies the remaining 6 of the 7 GTSs. While LN2 is transmitting data, MN4 continuously sends GTS allocation requests in an attempt to occupy the last GTS. Once LN2 releases its GTS at the 50-second mark, the coordinator allows MN4 to occupy the last GTS. MN4 now stops sending GTS allocation requests to conserve energy. LN2 sends a GTS allocation request around the 60-second mark and the 90-second mark, but the coordinator does not assign LN2 a GTS (because MN4 has them all). To see another iteration of this, we turn off the PAN coordinator around the 130-second mark to force it to perform the preventative CAP maintenance action manually (this is because the IEEE 802.15.4 source code from the Open-ZB does not handle this situation as it should). Accordingly, the PAN coordinator does not have any requested GTSs. Around the 140-second mark, we turn on the PAN coordinator and LN2 successfully is allocated one GTS and it transmits data during the corresponding CFP for about 70 seconds. MN4 now begins sending GTS allocation requests between the 150-second mark and 200-second mark and is able to occupy 6 GTSs. Also, when LN2 releases its GTS around the 200-second mark, MN4 immediately occupies all 7 GTSs again.

7.5 Stealing Network Bandwidth

Figure 7.4 (b) shows the data throughputs of LN2 and MN4 and the GTS allocation requests of MN4. While LN2 has one GTS and transmits data during the CFP, MN4 starts sending GTS allocation requests with 7 forged IDs around 20-second mark and transmits data at the assigned GTSs. One of 7 GTS allocation requests of MN4 is discarded at the first attempt because one GTS is already assigned to LN2. However, as soon as LN2 releases its GTS around the 50-second mark, MN4 occupies the last GTS immediately and has all

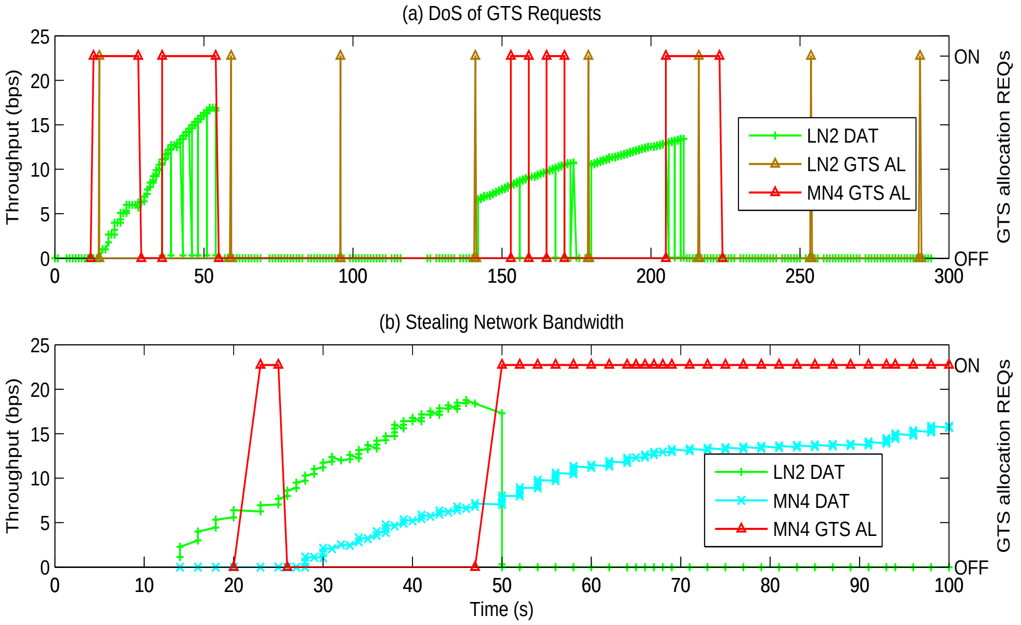


Figure 7.4. A malicious node (MN4) filling up all 7 GTSs. LN2 DAT: LN2 Data, LN2 GTS AL: LN2 GTS allocation request, LN2 GTS DE: LN2 GTS deallocation request, and MN4 GTS AL: GTS allocation requests from MN4. A malicious node (MN4) stealing GTSs during CFP. LN2 DAT and MN4 DAT: Data from LN2 and MN4 respectively and MN4 GTS: GTS allocation requests from MN4.

7 GTSs. As a result, LN2 and the PAN coordinator will use a lot of energy because LN2 continues to send GTS allocation requests to secure a GTS, and the PAN coordinator will continue to receive the illegitimate data from MN4.

Chapter 8

SECURING BEACON-ENABLED 802.15.4

In this chapter, we discuss which security requirements are necessary to defend against attacks on beacon-enabled 802.15.4 networks. As we demonstrated, attacks can be conducted by impersonating both the PAN coordinator and legitimate regular nodes. The legitimate regular nodes trusted forged beacons from a malicious node masquerading as the PAN coordinator (e.g., synchronization attack and DoS of data transmission by impersonating the PAN coordinator). The PAN coordinator also trusted forged data and control (GTS (de)allocation requests) packets from malicious nodes masquerading as legitimate regular nodes (e.g., DoS of data transmission by impersonating a legitimate node, DoS of GTS requests, false data injection, and stealing network bandwidth).

To address the aforementioned attacks, we propose an enhanced MiniSec [9], called beacon-enabled MiniSec (*BCN-MiniSec*), which satisfies the security requirements shown in Table 8.1. BCN-MiniSec can guarantee unicast authentication and broadcast authentication together in the beacon-enabled 802.15.4 network. Unicast authentication is used for the PAN coordinator to verify data and control messages. Given that MiniSec has low communication cost for unicast data message authentication, we couple the techniques used by MiniSec with lightweight authentication for unicast control messages in 802.15.4 beacon-enabled mode. With unicast data message authentication, BCN-MiniSec can prevent a malicious node from transmitting forged data messages. Since forged data messages are sent particularly at GTSs for false data injection, it is also possible that BCN-MiniSec verifies control messages (GTS allocation requests) for indirect defense beforehand. Primarily, BCN-MiniSec adopts unicast control message authentication to defend against DoS of data transmission by impersonating a legitimate node, DoS of GTS requests, and stealing network bandwidth that exploit forged GTS (de)allocation requests. Further, we add broadcast authentication to secure the PAN coordinator's broadcast messages. BCN-MiniSec adopts a one-way key chain for broadcast authentication as done in other works [17] [36] and discloses a key of the key chain immediately with the beacons for real-time communications in the beacon-enabled 802.15.4 network. We explain each security requirement below to show how the individual requirement works in the beacon-enabled 802.15.4 network and can defend against the attacks.

Table 8.1. Security requirements to counter the attacks.

Attacks		Security requirements	
		Broadcast authentication	Unicast authentication
Synchronization attack		Y	N
DoS of data transmission	Impersonating a legitimate node	N	Y
	Impersonating the PAN coordinator	Y	N
False data injection		N	Y
DoS of GTS requests		N	Y
Stealing network bandwidth		N	Y

8.1 Unicast control message authentication

In several of our attacks, a malicious node is able to send GTS (de)allocation requests with forged IDs. This is possible because the PAN coordinator does not sufficiently authenticate control messages (GTS requests). To prevent these attacks, *unicast* command frames should be authenticated properly with a message authentication code (MAC) using a keyed cryptographic hash function by the PAN coordinator. While BCN-MiniSec keeps MiniSec's offset codebook (OCB) [37] to generates MACs and counters for unicast data messages, BCN-MiniSec instead adopts a cipher block chaining MAC (CBC-MAC) with the advanced encryption standard (AES) to provide authentication with a 4-byte MAC and 1-byte initialization vector for unicast control messages. A number of MAC algorithms have been proposed and adopted in many other communication protocols for lightweight authentication. One of them is the CBC-MAC that uses a block cipher to obtain a MAC for messages. Since the CBC-MAC is able to process arbitrary length messages due to the block chaining method and has reasonable performance for authentication with AES (due to a minimum number of cipher function calls [38]), it is sufficient for authentication of unicast control messages. The steps for applying AES-CBC-MAC are as follows:

Key setup and nonce: Each legitimate node has its own secret key shared with the PAN coordinator (K_{LNi}, where i is $\{1, 2, ..., n\}$, the index of each legitimate node (LN) in the PAN) before the nodes are deployed. For the nonce (that provides semantic security and freshness), legitimate nodes also have their own initialization vectors (IVs) that can be exposed either explicitly or implicitly (e.g., only part of the IV exposed) with the control messages to the PAN coordinator. For instance, LNi has IV_{LNi-t}, where t is a sequence of packets increasing in a timely fashion.

Transmitting control messages: Each legitimate node constructs its control messages and transmits them as follows:

$$\begin{aligned} LN \rightarrow & PAN\,COR : CMD_t \\ & = MHR_t,\ MSDU_t,\ IV_{LNi-t}, \\ & \quad MAC(K_{LNi},\ MHR_t\,|MSDU_t\,|\ IV_{LNi-t}\,|L_t) \end{aligned}$$

Where LN stands for legitimate node, $PANCOR$ stands for the PAN coordinator, and L represents the length of the packet. CMD_i is a command frame as shown in Figure 8.1 (b†). MHR_t represents MAC header and $MSDU_t$ represents MAC service data unit as shown in Figure 8.1.

Verifying control messages: The PAN coordinator decodes the MAC with K_{LNi} of LNi

and gets MHR_t and IV_{LNi-t}. Then, it checks the integrity of MHR_t and $MSDU_t$ and authenticates LNi by K_{LNi}. It also confirms the packet's freshness with IV_{LNi-t}.

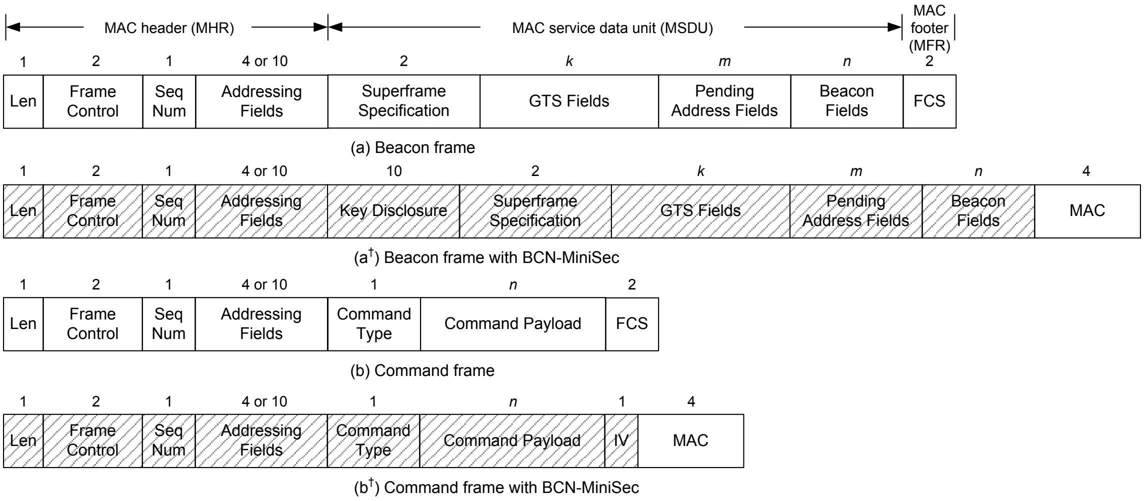


Figure 8.1. The packet formats of the 802.15.4 MAC with "no security" in (a) and (b) and BCN-MiniSec in ($a^\dagger$) and ($b^\dagger$). In ($a^\dagger$), Key disclosure represents a disclosed key in one-way key chains. IV is an initialization vector. The shaded regions represent authenticated fields.

8.2 Broadcast authentication

BCN-MiniSec also provides broadcast authentication for beacons in the beacon-enabled 802.15.4 network. While unicast authentication can provide the PAN coordinator with the trust of command and data frames that are transmitted from legitimate regular nodes, broadcast authentication is an efficient way to ensure the authenticity of the PAN coordinator's beacon broadcasts. Unlike unicast command and data frames sent by regular nodes to a single entity (i.e., the PAN coordinator), beacon frames are *broadcast* to all nodes in the PAN. During *broadcasts*, the nodes share the same key as the PAN coordinator in a similar fashion to unicast authentication. However, if this key is not dynamic, any legitimate node can masquerade as the PAN coordinator. For this reason, BCN-MiniSec uses one-way key chains [39] that use a one-way function to generate a sequence of keys. Many techniques similar to one-way key chains have been proposed for many cryptographic applications and particularly for broadcast authentication [17] [36]. The one-way key chains are generated by the PAN coordinator only, and the last key of the key chain is shared between the PAN coordinator and the regular nodes. Then, the PAN coordinator discloses the previous key in the key chain whenever it broadcasts beacons (i.e., one key per broadcast). The nodes can then authenticate the beacons.

Key setup: The PAN coordinator generates a key chain $(K_0, K_1, K_2, ..., K_{n-1}, K_n)$ from K_0 by using a cryptographic hash function F. Thus, $K_n = F\,(\,K_{n-1}\,)$, where n is the maximum number of keys (new keying material would be exchanged prior to $n = 1$ to continue the secure communications). Then, K_n (i.e., the last key) is pre-shared between the PAN coordinator and the nodes before they are deployed. K_n is used to secure the first beacon. Then, the next to the last key (K_{n-1}) is used with the second beacon. The pattern continues and the PAN coordinator traverses the chain backwards, using the previous key in the chain for the next transmission. The key expires after each transmission and the function that generates the key is one way so nodes cannot masquerade as the PAN coordinator.

Broadcast beacons: The PAN coordinator sends beacons as follows:

$$\begin{aligned} LN \leftarrow & PAN\,COR : BCN_{n-1} \\ & = MHR_{n-1},\ K_{n-1},\ MSDU_{n-1}, \\ & \quad MAC(K_{n-1},\ MHR_{n-1}\,|\,K_{n-1}\,|\,MSDU_{n-1}) \end{aligned}$$

Where $n-1$ is a sequence of the key chain and n is decremented by 1 after each transmission while $n > 0$. BCN_{n-1} is a beacon as shown in Figure 8.1 (a†).

Verifying beacon: The nodes verify K_{n-1} by using the hash function F. $K_n = F\,(\,K_{n-1}\,)$, where K_n is already verified in the previous beacon or trusted in the case of the initial key disclosed pre-deployment. That is, the nodes receive the previous key in the chain and assume that it is valid if its hash is the current key. Thus, using one-way key chains enables the PAN coordinator to efficiently provide authentication to its beacon broadcasts.

8.3 BCN-MiniSec Communication Costs

In this section, we estimate the additional communication cost of BCN-MiniSec. Figure 8.1 (a) and (b) show beacon and command frames of the 802.15.4 standard. Figure 8.1 ($a^\dagger$) and ($b^\dagger$) illustrate the frames using BCN-MiniSec to show the modified fields and the frame length. BCN-MiniSec adds *Key Disclosure and MAC* fields to regular beacons for broadcast authentication and *IV and MAC* fields to regular command frames for unicast control message authentication. In addition, BCN-MiniSec uses the same packet format for unicast data message authentication as that used by MiniSec.

Table 8.2 shows a summary of the communication costs of our countermeasure, BCN-MiniSec, with its viability. In the table, we present the packet sizes ($MHR, MSDU, MFR$) from our experiments (beacon frames, GTS command frames, and data frames). “No security” shows the packet sizes for each without any security applied and we compare the increased overhead (communication cost) and viability of MiniSec, BCN-MiniSec, and “No security.” In beacon-enabled 802.15.4 networks, BCN-MiniSec can prevent the synchronization attack and DoS of data transmission by impersonating the PAN coordinator with 31.6% overhead whereas MiniSec is not viable. Moreover, BCN-MiniSec requires 25% overhead to secure unicast control messages (GTS (de)allocation requests) whereas MiniSec is not viable. For unicast data message authentication, BCN-MiniSec requires the same 12.5% overhead as that of MiniSec since BCN-MiniSec uses the same techniques for data message authentication as those used by MiniSec. Thus, BCN-MiniSec is viable to defend against attacks on the beacon-enabled 802.15.4 MAC with reasonable overhead. It must be noted that the storage costs for the hash chain for broadcast authentication and keying material for unicast authentication have not been considered.

Table 8.2. The viability and communication cost of BCN-MiniSec. [1] MAC is message authentication code in this table.

Attacks	Security	Viability	Communication cost (B)					Increase over no security
			MHR	MSDU	MFR (+MAC[1])	Security overhead	Total	
Synchronization attack DoS of data transmission -. Impersonating the PAN coordinator	No security (beacon)	N	10	26	2	-	38	-
	MiniSec	N	-	-	-	-	-	-
	BCN-MiniSec	Y	10	36	4	12	50	31.6%
DoS of data transmission -. Impersonating a legitimate node DoS of GTS requests Stealing network bandwidth	No security (GTS request)	N	8	2	2	-	12	-
	MiniSec	N	-	-	-	-	-	-
	BCN-MiniSec	Y	8	3	4	3	15	25%
False data injection	No security (data)	N	12	2	2	-	16	-
	MiniSec	Y	12	2	4	2	18	12.5%
	BCN-MiniSec	Y	12	2	4	2	18	12.5%

Chapter 9

CONCLUSION AND FUTURE WORK

In this thesis, we first described the existing vulnerabilities of the beacon broadcast and the GTS management scheme in the IEEE 802.15.4 standard. We also investigated security protocols proposed in recent years and security mechanisms adopted in the standard. However, to date, no method comprehensively addresses the weakness of the beacon-enabled 802.15.4 MAC. To demonstrate the vulnerabilities in the 802.15.4 MAC, we implemented six attacks: (1) Synchronization attack, (2) DoS of data transmission by impersonating a legitimate node, (3) DoS of data transmission by impersonating the PAN coordinator, (4) False data injection, (5) DoS of GTS requests, and (6) Stealing network bandwidth. We also analyzed the results for each attack and designed a countermeasure, BCN-MiniSec. Future work will provide a detailed energy measurement of our attacks.

REFERENCES

[1] AlertMe home monitoring homepage. `http://www.alertme.com/products/home-monitoring`.

[2] A. Mishra, C. Na, and D. Rosenburgh, "On Scheduling Guaranteed Time Slots for Time Sensitive Transactions in IEEE 802.15.4 Networks," in *In Proceedings of Military Communications Conference*, 2007.

[3] A. Koubaa, M. Alves, and E. Tovar, "i-GAME: an implicit GTS allocation mechanism in IEEE 802.15.4 for time-sensitive wireless sensor networks," in *Proceedings of the 18th Euromicro Conference on Real-Time Systems*, 2006.

[4] A. Koubaa, M. Alves, and E. Tovar, "GTS allocation analysis in IEEE 802.15.4 for real-time wireless sensor networks," in *Proceeding of the 20th International Parallel and Distributed Processing Symposium*, 2006.

[5] F. Chen, T. Talanis, R. German, and F. Dressler, "Real-time enabled IEEE 802.15.4 sensor networks in industrial automation," in *Proceedings of the IEEE International Symposium on Industrial Embedded Systems*, 2009.

[6] P. Park, C. Fischione, and K. Johansson, "Performance Analysis of GTS Allocation in Beacon Enabled IEEE 802.15.4," in *Proceedings of the 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, 2009.

[7] A. Mehta, G. Bhatti, Z. Sahinoglu, R. Viswanathan, and J. Zhang, "Performance analysis of beacon-enabled IEEE 802.15.4 MAC for emergency response applications," in *Proceedings of the 3rd International Symposium on Advanced Networks and Telecommunication Systems (ANTS)*, 2009.

[8] J. R. Douceur, "The Sybil Attack," in *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS'02)*, (New York, NY, USA), IPTPS, 2002.

[9] M. Luk, G. Mezzour, A. Perrig, and V. Gligor, "MiniSec: a secure sensor network communication architecture," in *Proceedings of the 6th international conference on Information processing in sensor networks (IPSN '07)*, ACM, 2007.

[10] M. Demirbas and Y. Song, "An RSSI-based scheme for sybil attack detection in wireless sensor networks," in *Proceedings of the International Symposium on World of Wireless Mobile and Multimedia Networks*, 2006.

[11] J. Yang, Y. Chen, and W. Trappe, "Detecting sybil attacks in wireless and sensor networks using cluster analysis," in *Proceedings of the 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, 2008.

[12] Q. Zhang, P. Wang, D. Reeves, and P. Ning, “Defending against Sybil attacks in sensor networks,” in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops*, 2005.

[13] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, “A pairwise key pre-distribution scheme for wireless sensor networks,” in *Proceedings of the 10th ACM conference on Computer and Communications Security*, ACM, 2003.

[14] D. Liu and P. Ning, “Establishing pairwise keys in distributed sensor networks,” in *Proceedings of the 10th ACM conference on Computer and Communications Security*, ACM, 2003.

[15] W. Du, J. Deng, Y. Han, S. Chen, and P. Varshney, “A key management scheme for wireless sensor networks using deployment knowledge,” in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2004)*, 2004.

[16] L. Eschenauer and V. D. Gligor, “A key-management scheme for distributed sensor networks,” in *Proceedings of the 9th ACM conference on Computer and Communications Security*, ACM, 2002.

[17] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, “SPINS: security protocols for sensor networks,” in *Proceedings of the 7th annual international conference on Mobile computing and networking (MobiCom ’01)*, ACM, 2001.

[18] C. Karlof, N. Sastry, and D. Wagner, “Tinysec: a link layer security architecture for wireless sensor networks,” in *Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys ’04)*, ACM, 2004.

[19] F. Amini, J. Misic, and H. Pourreza, “Detection of Sybil Attack in Beacon Enabled IEEE802.15.4 Networks,” in *Proceedings of the International Wireless Communications and Mobile Computing Conference*, 2008.

[20] “Wireless medium access control and physical layer specications for low-rate wireless personal area networks. IEEE Standard, 802.15.4-2003,” May 2003. ISBN 0-7381-3677-5.

[21] N. Sastry and D. Wagner, “Security considerations for IEEE 802.15.4 networks,” in *Proceedings of the 3rd ACM workshop on Wireless security (WiSe ’04)*, ACM, 2004.

[22] M. A. Alim and B. Sarikaya, “EAP-Sens: a security architecture for wireless sensor networks,” in *Proceedings of the 4th Annual International Conference on Wireless Internet (WICON ’08)*, (ICST, Brussels, Belgium, Belgium), ICST, 2008.

[23] L. B. B. Aboba, J. C. J. Vollbrecht, and H. Levkowetz, “Extensible Authentication Protocol EAP.” `http://tools.ietf.org/html/rfc3748`, June 2004.

[24] T. Clancy and H. Tschofenig, "Extensible Authentication Protocol - Generalized Pre-Shared Key EAP-GPSK method." `http://tools.ietf.org/html/rfc5433`, February 2009.

[25] O. D. Radosveta Sokullu and I. Korkmaz, "On the IEEE 802.15.4 MAC Layer Attacks: GTS Attack," in *Proceedings of Second International Conference on Sensor Technologies and Applications (SENSORCOMM '08)*, 2008.

[26] T. Roosta, S. Shieh, and S. Sastry, "Taxonomy of Security Attacks in Sensor Networks and Countermeasures," in *In Proceedings of the First IEEE International Conference on System Integration and Reliability Improvements*, 2006.

[27] A. Wood and J. Stankovic, "Denial of service in sensor networks," *Computer*, vol. 35, pp. 54–62, Oct 2002.

[28] S. S. Jung, M. Valero, A. Bourgeois, and R. Beyah, "Attacking Beacon-Enabled 802.15.4 Networks," in *Proceedings of the 6th International ICST Conference on Security and Privacy in Communication Networks*, (Berlin, Heidelberg), Springer Berlin Heidelberg, 2010.

[29] Moteiv Corporation, *tmote-sky-datasheet*, 2006.

[30] Chipcon Products from Texas Instruments, *User Manual Rev. 1.0 CC2420DK Development Kit.*

[31] Texas Instruments Incorporated, *SmartRF Packet Sniffer User Manual Rev. 1.9.*

[32] Open-zb homepage. `http://www.open-zb.net/`.

[33] TinyOS homepage. `http://www.tinyos.net/`.

[34] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," in *Proceedings of the First IEEE. 2003 IEEE International Workshop on Sensor Network Protocols and Applications*, 2003.

[35] J. A. S. Anthony D. Wood, *A Taxonomy for Denial-of-Service Attacks in Wireless Sensor Networks.* CRC Press, 2004.

[36] S. Zhu, S. Setia, and S. Jajodia, "LEAP: Efficient security mechanisms for large-scale distributed sensor networks," in *Proceedings of the 10th ACM conference on Computer and Communications Security*, ACM, 2003.

[37] P. Rogaway, M. Bellare, and J. Black, "OCB: A block-cipher mode of operation for efficient authenticated encryption," *ACM Trans. Inf. Syst. Secur.*, vol. 6, no. 3, pp. 365–403, 2003.

[38] H. H. S. Frankel, "The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec." `http://www.faqs.org/rfcs/rfc3566.html`, September 2003.

[39] L. Lamport, "Password authentication with insecure communication," *Commun. ACM*, vol. 24, pp. 770–772, November 1981.