Anomaly Detection in Beacon-Enabled IEEE 802.15.4 Wireless Sensor Networks

Eirini Karapistoli and Anastasios A. Economides

Department of Information Systems,
University of Macedonia
Egnatia 156, Thessaloniki, Greece
{ikarapis,economid}@uom.gr

Abstract. During the past decade, wireless sensor networks (WSNs) have evolved as an important wireless networking technology attracting the attention of the scientific community. With WSNs being envisioned to support applications requiring little to no human attendance, however, these networks also lured the attention of various sophisticated attackers. Today, the number of attacks to which WSNs are susceptible is constantly increasing. Although many anomaly detection algorithms have been developed since then to defend against them, not all of them are tailored to the IEEE 802.15.4 standard, a dominant communication standard for low power and low data rate WSNs. This paper proposes a novel anomaly detection algorithm aimed at securing the beacon-enabled mode of the IEEE 802.15.4 MAC protocol. The performance of the proposed algorithm in identifying intrusions using a rule-based detection technique is studied via simulations.

Keywords: Wireless Sensor Networks, Beacon-enabled IEEE 802.15.4 MAC, Rule-based Anomaly Detection.

1 Introduction

WSNs raise the interest of different business domains, including that of security [1]. Their ability to monitor and control physical environments and large scale critical infrastructures make them a promising candidate. WSNs can be relatively easily deployed in a large geographical span, and can provide with fault diagnosis, intrusion detection and monitoring services in a cost-efficient manner since they do not require additional infrastructure. While the distributed nature of a WSN increases the survivability of the network in critical situations (it is much less likely that the network will be affected in its entirety by failures or attacks), defensive mechanisms that could protect and guarantee the normal operation of the WSN in the presence of adversaries are still needed.

Currently, research on providing security solutions for WSNs has mainly focused in key management [2], [3], secure authentication and routing [4], secure localization and data aggregation [5], [6], and recently, in intrusion detection [7].

T. Zia et al. (Eds.): SecureComm 2013, LNICST 127, pp. 1–18, 2013.

Within the limited scope of this paper, we restrain our focus on the latter approach in an attempt to defend against strong inside attackers that have penetrated the first perimeter of defense. An Intrusion Detection System (IDS) monitors the events occurring in the network and analyzes them to detect signs of intrusion [8]. Various signature-based and anomaly-based IDS architectures have been proposed for flat and hierarchical WSNs [9], [10], [11]. However, to the best of our knowledge, none of them is applicable to IEEE 802.15.4-compliant WSNs.

The IEEE 802.15.4-2011 standard [12] is a dominant communication standard developed to provide low-power and highly reliable wireless connectivity among inexpensive, battery-powered devices. While emphasis has been given on improving the performance of the 802.15.4 MAC protocol [13], limited work has been contacted on securing its *beacon-enabled* mode. As identified in [14], [15], this mode is vulnerable to a number of attacks. Some of the attacks (i.e., radio jamming and link layer jamming) are common to all MAC layer definitions. Others like the back-off manipulation and the attacks against the acknowledgement mechanism may also occur in IEEE 802.11 wireless networks due to some common properties in the MAC layer implementations [16]. However, several attacks including the Personal Area Network (PAN) identifier conflict attack, and the Guaranteed Time Slot (GTS) attack are only applicable to the 802.15.4 MAC layer mechanisms defined by the standard. Therefore, the latter category of attacks requires novel, anomaly-based intrusion detection algorithms to defend against them.

Accordingly, this work contributes to the area of anomaly detection for IEEE 802.15.4-compliant wireless sensor networks. We propose a distributed anomaly detection algorithm for securing the beacon-enabled mode of the IEEE 802.15.4 MAC protocol. Vulnerabilities of the underlying MAC are exposed and dealt with using a rule-based detection approach. Our algorithm differentiates from existing works in that it does not rely on the existence of special types of nodes, i.e. monitor nodes or watchdogs, to perform the anomaly detection task. Finally, the proposed algorithm does not require expensive communication between the sensor nodes, since anomaly detection and revocation are performed distributively.

The remainder of the paper is organized as follows: in Section 2, existing work on securing the beacon-enabled mode of the 802.15.4 MAC is outlined. In Section 3, we review several features of the underlying MAC protocol and analyze its vulnerabilities in order to provide a better understanding of the proposed algorithm. In Section 4, we provide a detailed description of our anomaly detection algorithm. Section 5 illustrates the obtained simulation results, followed by detailed reports. Finally, conclusions are given in Section 6.

2 Related Work

Several defensive methods have been proposed for securing the beacon-enabled mode of the IEEE 802.15.4 MAC protocol. The standard itself encompasses

built-in security features to provide data secrecy and data authenticity. However, as Sastry *et al.* [17] pointed out, these security features have vulnerabilities related to the initial vector (IV) management, key management, and integrity protection. To address these issues, Alim *et al.* introduced EAP-Sens [18], a link layer secure protocol implementation for 802.15.4 sensor networks in beacon-enabled mode. While effective in its design, as with any authentication protocol, EAP-Sens is vulnerable to insider attacks launched by compromised (malicious) nodes.

Sokullu *et al.* [14], [19] were the first to analyze insider attacks targeting the beacon-enabled mode of the IEEE 802.15.4 MAC protocol. The authors used ns2 simulations to demonstrate DoS-like GTS attacks whose main goal was to create collisions at the GTS slots and deny the guaranteed communication. While effective, the major drawback of their work is that it lacks a clear description of how to defend against such attacks. Amini *et al.* [20] proposed a Received Signal Strength Indicator (RSSI)-based solution to detect Sybil attacks in IEEE 802.15.4 beacon-enabled clusters. The coordinator is tasked with detecting anomalies inside its cluster based on deviations in the tuple (disc number, device ID) it assigned to its cluster members. However, this method does not consider the case of compromised coordinators. Moreover, if a malicious node is close enough to a legitimate node, its RSSI may be confused with the RSSI of the legitimate node, thus enabling the malicious node to escape detection. Recently, Jung *et al.* [15] performed an in-depth study of the vulnerable properties of the beacon-enabled mode of the IEEE 802.15.4 standard. The authors implemented on real devices four potential insider attacks associated with those vulnerabilities, and presented mechanisms to defend against them. While the authors provide a good framework to analyzing IEEE 802.15.4 MAC layer attacks, no implementation or testing exists relative to the defensive mechanisms they propose in their paper.

Overall, a concrete framework for securing IEEE 802.15.4-compliant sensor networks against insider attacks is still missing. Our approach to the problem is to use rule-based anomaly detection. As analyzed in [11], rule-based anomaly detection is attractive because its methodology is flexible and resource-friendly and benefits from the absence of an explicit training procedure. In rule-based detection, the anomaly detector uses predefined rules to classify data points as anomalies or normalities. While monitoring the network, these rules are selected appropriately and applied to the monitored data. If the rules defining an anomalous condition are satisfied, an anomaly is declared. Da Silva *et al.* [21] where among the first to propose a rule-based distributed ADS for WSNs. While the authors provide a good framework to rule-based detection, the defined rules are not applicable to attacks targeting beacon-enabled WSNs. This is also the case for other similar rule-based anomaly detection systems (ADS) proposed for WSNs [22], [23].

Therefore, in this paper, we attempt to move towards that direction proposing a specific modular rule-based ADS architecture tailored to IEEE 802.15.4-compliant wireless sensor networks operating under the beacon-enabled mode.

3 Preliminaries

3.1 The IEEE 802.15.4 MAC

The MAC sublayer of the standard provides services such as beacon generation and synchronization, PAN association and disassociation, GTS management, and channel access among others [12]. It also provides support for star and peer-to-peer network topologies. Peer-to-peer topologies allow more complex network formations to be implemented, such as mesh and cluster tree topologies that are better suited for security-oriented applications. When lines of communication exceed the implementation-specific personal operating space (POS), an 802.15.4 network can be a self-configuring, multi-hop network. Two different device types can participate in an IEEE 802.15.4 network; a full-function device (FFD) and a reduced-function device (RFD). An FFD device can operate in three modes serving as a PAN coordinator, a coordinator, or a device. An RFD device instead, only connects to a cluster tree network as a leaf device at the end of a branch (see Fig.1).

The IEEE 802.15.4 MAC operates in two modes, the *beacon-enabled* mode and the *nonbeacon-enabled* mode. In the beaconless mode, coordinators do not emit regular beacons. Moreover, channel access is managed through the unslotted version of the CSMA-CA algorithm. In beacon-enabled PANs, which can assume only a star or tree topology, all non-leaf nodes periodically transmit beacon frames. In this mode, a PAN coordinator relies on a superframe (SF) structure to enable transmission and reception of message that consists of a beacon, an active period, and an inactive period. While starting a PAN, coordinator sets its *macPANId* and the length of both active and inactive periods, defined by the *macBeaconOrder*, BO=[0,15), and the *macSuperframeOrder*, SO=[0,BO), respectively. The active period consists of 16 equal sized time slots and contains a contention access period (CAP), which uses slotted CSMA/CA for channel access, and a contention free period (CFP), which consists of guaranteed time slots (GTS) that are allocated on demand to nodes for a contention-free access to the channel. Member nodes can switch over to sleep mode during the inactive period to save battery.

3.2 Attacking the IEEE 802.15.4 MAC

The beacon-enabled mode of the IEEE 802.15.4 MAC protocol is vulnerable to a number of internal and external attacks several of which are common to all wireless MAC layer definitions. Therefore, in this paper we concentrate on attacks that target peculiar mechanisms of the underlying MAC, namely its PANID conflict resolution procedure, the GTS allocation and deallocation mechanisms and the data transmissions during the CAP and CFP portions of the superframe.

PANId Conflict Attack. According to the IEEE 802.15.4 standard [12], the PAN identifier conflict resolution procedure is executed when more than one PAN coordinators with the same PANId operate in the same POS. If such a conflict

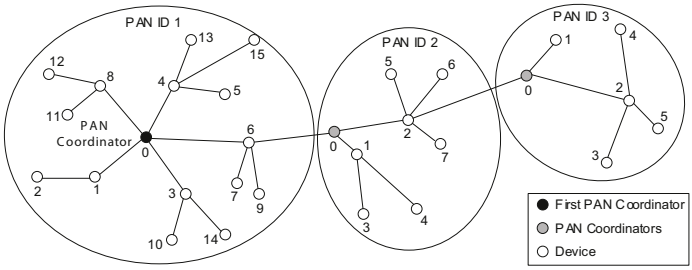

Fig. 1. An IEEE 802.15.4 cluster tree network. (Source: [12]).

occurs, a member device that receives beacons from both PAN coordinators, can notify its PAN coordinator to perform the conflict resolution procedure. An adversary device can take advantage of this vulnerability and frequently send fake PANId conflict notification commands to the coordinator and oblige the latter to perform the conflict resolution procedure. Such an attack may prevent or greatly delay communication between devices and the PAN coordinator.

GTS Attack. According to [12], a GTS slot is the portion of the superframe that provides contention free communication between a device that reserved the slot and a coordinator. The GTS allocation mechanism is executed as follows. First, the device has to receive the beacon frame to identify the superframe boundaries. A GTS allocation request is sent in the CAP portion of the superframe to the coordinator. The request includes the required length and direction (uplink or downlink) of the GTS slot. The coordinator may send an ACK packet to confirm the successful reception of the GTS request. If GTS slots are available, the coordinator assigns them to the requesting device using the beacon frame. Once assigned, the data transmission takes place in the GTS slots of the following superframes. Similarly, a deallocation request results to the deallocation of a GST slot.

As it can be seen, the GTS management scheme does not verify the ID of each device that requests GTS allocation or deallocation. Therefore, an inside attacker can easily compromise this procedure by either impersonating existing legitimate nodes' IDs or creating new IDs for devices that do not exist (i.e., implement a Sybil attack at the MAC layer [24]). Let us examine the possible attack scenarios separately. In the first attack scenario, a malicious node that is in the POS of the PAN coordinator first obtains the IDs of existing legitimate nodes in the PAN by either overhearing the list of pending addresses in the beacon frame or the GTS allocation requests that are sent during the CAP. Accordingly, when a legitimate node requests GTS allocation to transmit data in the CFP portion of the next superframes, the malicious node can cancel this transmission by sending a GTS deallocation request using the spoofed ID immediately after the GTS allocation request as shown in Fig 2. Since the PAN coordinator receives the deallocation request while processing the GTS allocation from the legitimate

node, it ignores the GTS allocation coming first and does not assign any GTS to the legitimate node. As a result, the legitimate node is not assigned any GTS and cannot transmit its sensed data.

In the previous attack, a malicious node impersonates a legitimate node to cause the PAN coordinator to deallocate the requested GTSs. In this attack, a malicious node sends GTS requests from multiple fake IDs (up to 7) to completely allocate the CFP period. To perform this attack, a malicious node continuously monitors the available GTS slots with the intent of completely occupying them. Then, the attacker sends GTS allocation requests to fill up all the available GTSs in the superframe (see Fig. 3 for an explanation). By occupying the available GTSs and not allowing legitimate nodes to reserve GTSs the malicious node performs an exhaustion and unfairness type of attack. The malicious node does not necessarily need to send data at the assigned time slots. However, occasionally it may need to do so in order to prevent the PAN coordinator from dropping the assigned GTSs.

False Data Injection. In this attack, the malicious node first identifies which legitimate node has not requested GTS allocation by looking at the GTS descriptors of the beacon frames. Then, it 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 uses the spoofed ID and sends false data to the PAN coordinator during the CFP. The legitimate node at the same time sends its sensed data during the CAP. After checking the node's ID, the PAN coordinator regards the false data as time-sensitive ones and ignores the data sent from the legitimate node during the CAP.

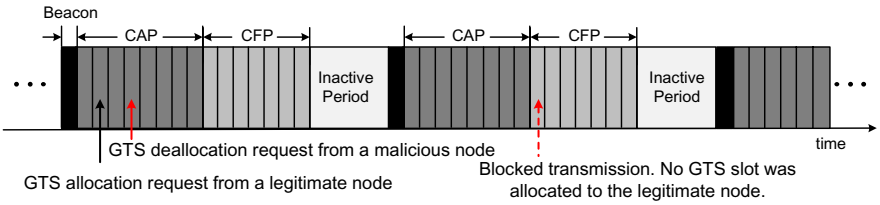

Fig. 2. A malicious node launching a GTS deallocation attack

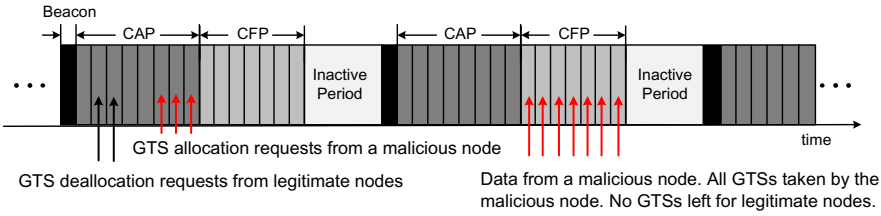

Fig. 3. A malicious node stealing all 7 GTSs of the CFP period

DoS-like Attacks. In this attack, the attacker has the ability to create collisions by jamming the beacons or specific GTS slots, which are broadcast in nature. In order to jam the beacons the malicious node must align to the superframe boundary and produce a collision by sending data at the start of the beacon. In the second case, it may intercept the beacons and learn in which GTS slots legitimate nodes send data. Then, it can corrupt the guaranteed communication between this device and the coordinator by jamming one or multiple GTSs.

Selective Forwarding and Black Hole Attacks. The communication flow in IEEE 802.15.4 beacon-enabled PANs, allows a captured (compromised) coordinator to perform a selective forwarding attack. In this attack, the malicious nodes refuses to forward all or a subset of the messages it receives from its child devices and simply drops them. If the attacker drops all the packets, the attack is then called black hole.

4 Anomaly Detection in 802.15.4-Based WSNs

This section highlights our anomaly detection framework, stating assumptions, and describing the proposed algorithm.

4.1 Assumptions of the Model

A number of assumptions are made concerning the framework in which the wireless sensor nodes operate. First, we consider a cluster tree 802.15.4 network in which most devices are FFDs. We assume that there is no pre-existing distributed trust model or peer-to-peer trust model, and hence no node can be fully trusted. Sensor nodes comprising the WSN remain stationary all the time. Once the clusters are formed and nodes are assigned short addresses, they maintain the same members, except for cases where nodes are blacklisted, die, or when new nodes join the network. Each node shall maintain a data structure that facilitates the storage of direct observations of all its *parent-child* nodes. Moreover, since sensor nodes are "weak" devices, we assume that an adversary can completely take over nodes and extract their cryptographic keys or load malicious software to launch an insider attack. Accordingly, and in order to limit the complexity of our model, we do not implement any cryptographic security mechanism, even though the MAC sublayer of the standard provides hooks that can be harnessed by upper layers to achieve authentication, message integrity, confidentiality and replay protection [17]. Next, we describe our anomaly detection algorithm in detail.

4.2 Detailed Algorithm Description

The core of the proposed algorithm relies on the periodic normal/guarding operation of the nodes comprising the WSN. To implement the aforementioned dual

behavior, nodes inside the network adopt a *periodic operation*. Each node establishes the periods of normal/guarding operation during the *cluster formation* and *guarding initialization* phases. After the clusters are formed and guards are assigned, the monitor node collects statistics for its peers, which are used during the *anomaly detection and node revocation* phase to detect signs of intrusion. The different phases of our algorithm are analyzed below.

Phase 1: Cluster Formation. In the proposed algorithm, sensor nodes follow the association procedure defined by the standard in order to gradually connect and form a multicluster network structure. Before starting a PAN, the first action a device needs to perform is to initiate an active or passive scan in order to locate other PANs within its POS. Once a new PAN is established, the PAN coordinator is ready to accept requests from other devices to join the PAN. In the process of joining a PAN, the device requesting association will perform a passive scan to determine which PANs in its POS are allowing association. A device should attempt to associate only with a PAN through a coordinator that is currently allowing association (i.e., a coordinator whose *macAssociationPermit* is set to TRUE). In order to impose topological restrictions on the formation of the network, the *macAssociationPermit* is set to FALSE when the number of nodes joining a particular PAN exceeds the parameter N_u. If the original candidate device is not able to join the network at that coordinator, it will search for another parent device or it will become the PAN coordinator of a new PAN adjacent to the first one by selecting a suitable PAN identifier (see Fig. 1). Every device follows this association procedure and gradually connects to a PAN.

After the clusters are formed, each node starts operating in one of the available two modes; the *normal mode* in which it collects and forwards application-specific sensor measurements to the base station (BS), and the *guarding mode* in which it promiscuously listens to its peers' transmissions in order to detect signs of intrusion. During the normal mode, nodes may exchange data in the active portion of the superframe. Three types of data transfer transactions are allowed in the IEEE 802.15.4 MAC. The first one is the *direct transmission* in which a device sends data to a coordinator. The second data transfer model is the *indirect transmission* in which a coordinator sends data to a device, and the third transaction is the *peer-to-peer* data transfer (see Fig. 4 for an explanation). Within our algorithm, nodes are allowed to perform direct transmissions in both the CAP and CFP portions of the superframe resembling a sink-based reporting scheme that is typical in WSNs. Each device shall transmit a data frame following the successful application of the slotted version of the CSMA-CA algorithm. The transmission procedure, which includes the acknowledgement mechanism, begins with a randomly selected back-off time. Any transmission procedure can be repeated (attempted), if it can be completed within the same portion of the superframe. The remaining data, if any, will be deferred to the next active portion of the superframe.

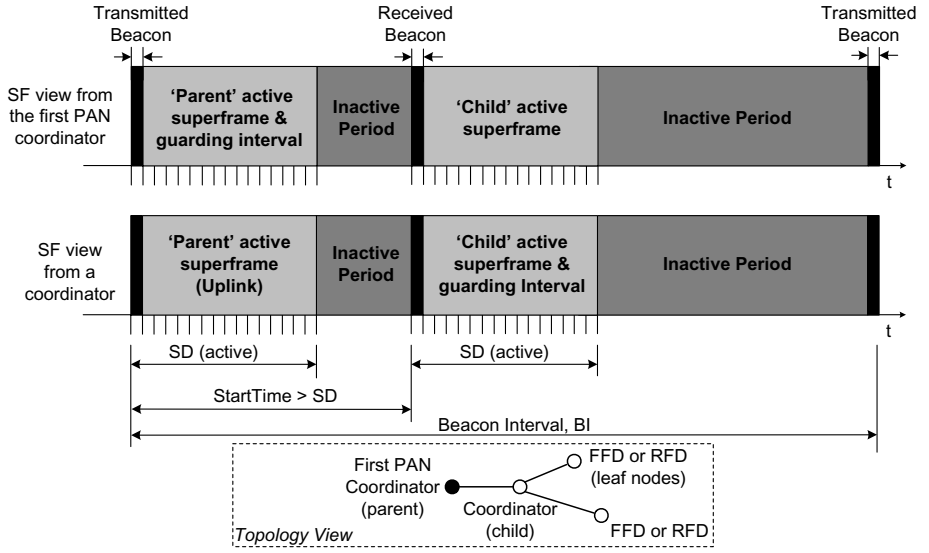

Fig. 4. Periodic normal/guarding operation of the nodes inside a PAN

Phase 2: Guarding Initialization. After completion of phase I, nodes periodically enter the guarding phase by enabling the *macPromiscuousMode* of the standard for a time equal to a *superframe duration*, SD. Time SD determines the active portion of the superframe in symbols, and relates to the *macSuperframeOrder*, $0 \leq \text{SO} \leq \text{BO} \leq 14$, as follows:

$$SD = aBaseSuperframeDuration * 2^{SO} symbols \quad (1)$$

where *aBaseSuperframeDuration*=960, and BO is the interval at which the coordinator shall transmit its beacon frames. During this time, also called the guarding interval, each node gathers traffic-related attributes for all its parent and child nodes (if any) by promiscuously listening to the packets transmitted over the shared communication channel (see Fig. 4). As it can be seen, a functional difference between the first PAN coordinator and the rest PAN coordinators is that the latter alternatively acts as an associated device (during the active period of the superframe of the first PAN coordinator) and as the coordinator (guard) of a set of surrounding FFDs or RFDs.

So, a node in the guarding mode is in charge of monitoring its parent-child nodes by turning the promiscuous listening mode on or equivalently by setting the *macPromiscuousMode* to TRUE. When in promiscuous mode, the MAC sublayer shall pass all frames correctly received to the next higher layer for further processing. Note that each guarding period is a unique guarding round for collecting traffic-related attributes. These attributes may then be used by the ADS system running on each sensor node to detect signs of intrusions. At this point we should also state that the guarding periods of the nodes inside

a PAN are not synchronized. Each cluster member enters the guarding mode sequentially and in accordance to its allocated *macShortAddress*. This is because we want to distribute the role of the guard among the cluster members and enable the detection and revocation to be fully distributed.

As apparent, the guarding periods are bounded by the beacons the PAN coordinators sent. Child devices associated with them use these beacons to synchronize their guarding periods. A guarding period is actually considered active following the transmission of the beacon frame. The beacon frames contain essential parameters of the PAN, such as the *CoordPANId*, the *macBeaconOrder*, the *macBSN*, and the *StartTime* at which the beacon frame was received. A node that receives a beacon frame stores its information locally and consequently learns the consecutive moments during which it will enter in the guarding mode. The reception of the beacon frame also confirms that the coordinator is still alive and operational, and that the device has not been orphaned. As apparent, keeping the correct timing for broadcasting/receiving these frames is the highest priority task for every node. Whatever a node is doing, i.e. is engaged in other transmit or receive operations, it will be interrupted for the accurate, on-time transmission/reception of the beacon frame. In order to acquire beacon synchronization and to maintain their periodicity, nodes need to set the *TrackBeacon* parameter to TRUE. This will enable a node to switch on its radio slightly before the expected broadcast of the beacon in order to receive it.

Phase 3: Anomaly Detection and Node Revocation. Our network-based ADS detects anomalies based on the packets that it monitors. Hence, following the data acquisition, anomaly detection and revocation come next. As already revealed, each node activates its ADS functionality when the MAC sublayer is in the so-called *promiscuous* (receive all) mode. During this guarding mode, each node keeps track of the transactions of all its parent-child nodes and stores the collected packet in a data structure. Since we follow a rule-based approach to anomaly detection, each data structure is evaluated according to the sequence of rules defined in Table 1. A packet is discarded after being tested against all rules without failing any of them. On the opposite case, an alarm will be raised if a violation of these rules occurs.

Indeed, an alarm indicates that a node is an intruder and needs to be revoked. Revocation is initiated following a process similar to the disassociation mechanism defined by the standard. Since nodes enter the guarding mode periodically, every node can independently verify intrusion instances and take revocation on the intruder. Note that revocation can be lazy, in that a node does not need to verify the intruder unless the latter is its parent or its child. In this way, attacks are detected and revoked in a fully distributed manner.

The MAC sublayer of the 802.15.4 standard allows us to implement the node revocation functionality easily since it defines procedures on how a device can disassociate from a PAN. The disassociation procedure may be initiated either from the PAN coordinator or from an associated device. Following the completion of a guarding period and the declaration of an anomaly, the coordinator

Table 1. Rules definition for detecting IEEE 802.15.4 MAC attacks

Rule Description	Attack Detected	Malicious actor
When a PANId conflict notification command is received at the coordinator, increase a counter. If after SD symbols, less than half of the associated devices report this conflict, raise an alarm.	PANId conflict attack	Associated device
If both GTS allocation & deallocation requests arrive from the same device in the CAP, raise an alarm. If an associated device does not send data during its allocated GTS slots for two consecutive CFP periods, raise an alarm.	GTS deallocation attack GTS allocation attack	Associated device Associated device
If data arrives in both the CAP and CFP portions of the same superframe, and originates from the same device, raise an alarm.	False data injection	Associated device
When half of the associated devices send an orphan notification request following the transmission of a beacon, raise an alarm.	DoS-like beacon attack	Associated device
If after two consecutive CFP periods, no packets are received during allocated GTS slots, raise an alarm.	DoS-like GTS attack	Associated device
When a packet is not forwarded as it should, increase a counter. When this counter reaches a threshold, t, after SD symbols, raise an alarm.	Selective Forwarding & Black hole attacks	Coordinator

may send a disassociate notification command to instruct a device to leave the PAN (*malicious associated device case*) or an associated device may request disassociation from the coordinator (*malicious coordinator case*). Let us examine these two cases separately.

When a coordinator wants one of its associated devices to leave the PAN, it sends a disassociation notification command to the malicious device. Because the disassociation command contains an acknowledgment request, the associated device shall confirm its receipt by sending an ACK frame. Even if the ACK is not received, the coordinator should consider the device disassociated. The next higher layer of a coordinator should disassociate a device by removing all references to that device. The device will soon conclude that it has been orphaned and will attempt to join other PANs within its POS. If no PANs exist in its POS, the revocation of the node would be global. In the opposite case, a similar procedure will be followed to revoke this node from the rest of the PANs it will attempt to associate with in the future.

In the second case, an associated device may send a disassociate notification command to notify the malicious coordinator of its intent to leave the PAN. Again, this command contains an ACK request. However, even if the ACK is not received, the device should consider itself disassociated. The orphaned device will in turn have to perform an active or passive scan in order to join other PANs that exist in its POS, or initiate a new PAN in case the active or passive scans fail. Since gradually all nodes associated with the malicious coordinator will independently verify the intrusion and leave its PAN, new clusters will be formed and the malicious coordinator will be completely revoked.

5 Performance Evaluation

5.1 Simulation Environment

In order to implement the proposed algorithm, we extended the capabilities of the existing IEEE 802.15.4 model developed in the OMNeT++ simulator [25]. This model was adapted from a version for ns-2 by Chen and Dressler. The model, which is described in more details in [26], implements the IEEE 802.15.4-2006 protocol stack. It also consists of two protocol-independent modules supporting energy measurement and mobility in the simulations. Our extension to the model targeted only the MAC sublayer (the PHY layer remained intact). Besides adding C++ code for the anomaly detection engine, modifications were made to the beacon-enabled mode of the model in order to support 802.15.4 cluster-tree topologies similar to [27]. In order to prevent overlapping, the emission of beacons is governed by an offset time. The offset for each PAN coordinator is set in a special *StartTime* parameter in the *omnetpp.ini* file. The particular value of the offset, which is null for the first PAN coordinator, for any other PAN coordinator is proportional to its *CoordPANId*.

We simulated a 802.15.4 cluster-tree network configured with our ADS. 20 nodes were placed uniformly at random in a rectangular playground of 100 x $100m^2$ (the first PAN coordinator (host[0]) was placed on the upper-left corner

of the network). Each node has a communication range (POS) of $20m$ and operates under the 2.4 GHz PHY. We set the maximum number of octets added by the MAC sublayer to the PSDU without security equal to *aMaxMPDUUnsecuredOverhead*, 25 octets. This leads to a DATA PPDU length of 31 bytes, a beacon PPDU length of 17 bytes and an ACK PPDU length of 11 bytes. Regarding the transceiver characteristics, we use those of the IEEE 802.15.4-compliant CC2420 Chipcon radio [28], where each sensor consumes as high as 19.7 mA, 17.4 mA and 20 μA, in receive, transmit and sleep modes respectively.

We simulated a security-oriented application supporting sink-based reporting, that is to say, traffic flowing from the devices to the BS (typical case of a sensor network). Since nodes perform upstream transmissions this fact guarantees that the packet will reach the BS. In this scenario, only leaf node were generating traffic. The traffic load was set equal to 2 packets per second. During the simulation, randomly selected intelligent adversaries include themselves in the network by replicating legitimate (captured) nodes. The malicious nodes selectively launch one of the attacks identified in Section 3.2. All the presented results were averaged over 10 simulation runs. Each run lasts for 20 minutes, which gives as an overall simulation time of 10 hours.

5.2 Simulation Results

In this section, we evaluate the performance of the proposed anomaly detection algorithm through simulations. Comparison of our algorithm with existing rule-based anomaly detection schemes would not be appropriate, as they are not tailored to the IEEE 802.15.4 standard. Two metrics were used to evaluate the effectiveness of our algorithm. These are the *percentage reduction in network lifetime*, which is used to examine the extent by which our ADS degrades the network lifetime when being implemented in common sensor nodes, and the *detection accuracy* defined as the ratio of the detected attacks to the total number of detected and undetected attacks.

Energy Consumption. Fig. 5a illustrates the percentage reduction in network lifetime as a function of the percentage increase in the number of compromised nodes. To simulate the described scenario, we chose at random a number of network nodes and we programmed them to selectively launch one of the attacks depicted in Table 1. With regard to selective forwarding attacks (launched only by non-leaf nodes), the attacker was dropping packets with a probability $p_d = 30\%$. When $p_d = 100\%$, the attacker was executing a black hole attack. We set the threshold value for the percentage of packets being dropped over the guarding interval, SD, to be $t = 20\%$. Above this threshold, an alarm was generated and node revocation was initiated. For all other types of attack, the counter-criterion rules of Table 1 are evaluated in succession and, if violated, an alarm is raised.

As the three curves show in Fig. 5a, the percentage reduction in network lifetime increases smoothly as the percentage of malicious nodes increases. This is because more energy-consuming intrusion detection functions are being executed

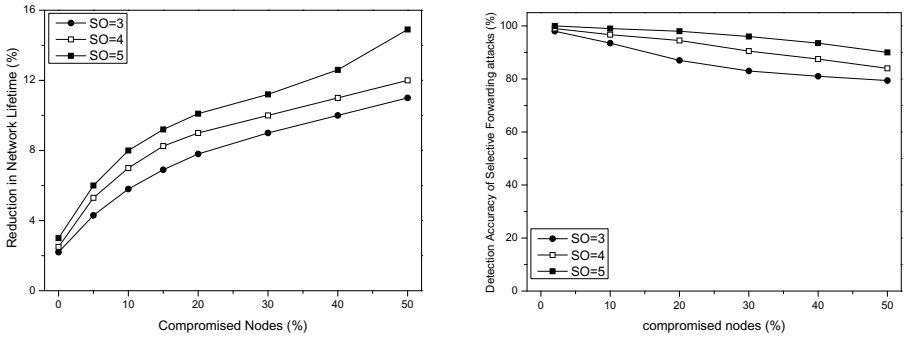

Fig. 5. a) Percentage reduction in network lifetime, and b) Detection accuracy of Selective forwarding attacks

following the introduction, identification and revocation of an increasing number of adversaries. Overall, the network lifetime decreases by as high as 14.8%. The relatively small decrease in network lifetime is achieved because the energy-consuming role of the guard is rotated among the network nodes, a fact that uniformly distributes the energy dissipation among the nodes. As expected, the percentage reduction in network lifetime is low when the network contains no malicious nodes. The obtained curves also indicate the trade off between the value of the *macSuperframeOrder*, SO={3,4,5} and the energy cost. According to Eq. (1), bigger SO values extend the guarding interval, SD, or equivalently the time window the monitor node is hearing in the promiscuous mode. As such, the longer a monitor node stays in the 'receive all' mode, the higher is the associated energy cost.

Detection Accuracy. The rest of the figures evaluate the effectiveness of our algorithm against the attacks depicted in Table 1. In each attack scenario, there was always one single type of attacker, which was varied in each simulation.

One interesting aspect these figures present is that the variation of the value of the SO does not impact the detection efficiency of our algorithm. Only the selective forwarding attacks and PANId conflict attacks, which are assessed over a time window SD, are affected by the value of SO. Indeed, as shown in Fig. 5b, smaller SO values result in lower detection levels. This happens because small SO values, result in small guarding intervals SD. Recall that the interval SD relates to the time window that a monitor node has in order to gather packets and analyze them for signs of intrusion. Since less packets are being collected as a result of the smaller SD interval, this affects the decision making process of anomaly detection and produces less accurate intrusion detection results. One aspect that is common in all types of attack is that if there is a high fraction of compromised nodes inside the network (50 percent or more), the detection levels achieved by our anomaly detection algorithm tend to drop below 90 percent,

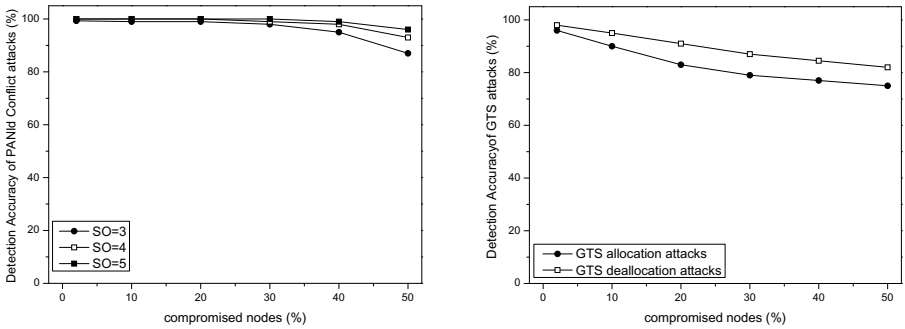

Fig. 6. Detection accuracy of a) PANId conflict attacks, and b) GTS attacks

which is considered low for an effective ADS. We can thus say that our ADS works acceptable when having 45% or less of compromised nodes inside the WSN.

As already revealed, Fig. 5b shows results on the detection accuracy of selective forwarding attacks. In this type of attack, since packets are dropped probabilistically, there might be the case that during the guarding interval of some nodes, the dropped packets are less than $t = 20\%$, and no alert is produced by those nodes. Then, the detection rule over that time window will not be satisfied, and would produce no alarm. This is less probable to happen if the value of SO gets bigger or if the nodes launch black hole attacks (the results on black hole attacks are not presented here due to space restrictions). In this case, the probability that during an SD interval the dropped packets are less than t, resulting in a false negative, is close to zero, and hence the accuracy in detecting that kind of attack is close to 100%.

Fig. 6a illustrates the detection accuracy of PANId conflict attacks. In PANId conflict attacks, detection was always close to 100% due to the rule being applied to detect this kind of attack. Another factor that keeps the detection levels high is that these attacks are not mistaken with any other kind of attack or with occasional network failures, and as such, a small number of false negatives is only generated. However, an increase in the number of misdetections is obtained when half, or more, of the nodes behave maliciously. In this case, the minority vote rule being applied does not prevail any more.

According to Fig. 6b, the detection of GTS attacks ranges between 99% and 80%. In this scenario, since the SO does not impact the detection levels, only results for SO=4 are depicted. The two curves indicate that the detection of GTS allocation attacks is less successful. This happens because this type of attack may be confused with the DoS-like GTS attacks, and as such, it may generate a higher number of false negatives.

Fig. 7a on the other hand, shows that false data injection attacks, similar to PANId conflict attacks, are detected with very high accuracy. Again, this attack is not mistaken with any other kind of attack or network failure, generating few false negatives.

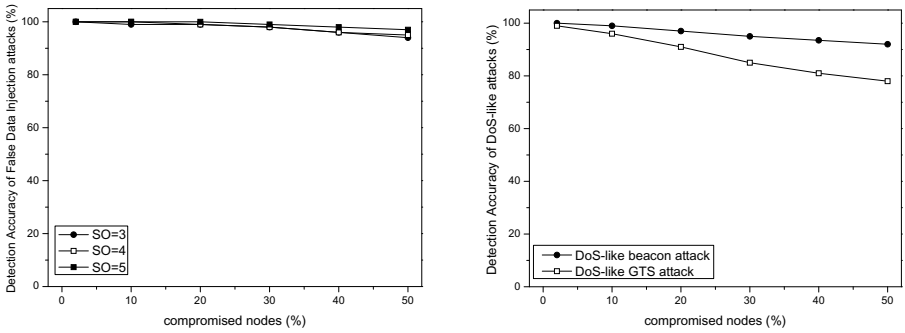

Fig. 7. Detection accuracy of a) False data injection attacks, and b) DoS-like attacks

Fig. 7b illustrates the obtained results on the detection accuracy of DoS-like attacks. While the DoS-like beacon attacks are detected with an accuracy always above 95%, the detection effectiveness drops in the case of DoS-like GTS attacks. This happens because in this type of attack there is no internal mechanism (similar to the lost synchronization) to notify the coordinator and assist the decision making process of anomaly detection. Moreover, these type of attacks may be confused with the GTS allocation attacks when jamming occurs in multiple GTS slots, a fact that may further increase the number of false negatives.

6 Conclusions and Future Work

In this paper, we presented a distributed anomaly detection algorithm for securing IEEE 802.15.4-compliant WSNs operating in the beacon-enabled mode. The proposed algorithm exploits the peculiar characteristics of the standard in order to incarnate the concept of periodic guarding for anomaly detection purposes. The OMNeT++ simulator has been used to implement our algorithm and to collect various results aiming at assessing its performance. The results showed that our approach maintains the energy consumption overhead at very low levels, while at the same time, it achieves high detection accuracy for all types of identified attacks. In the future, we intend to examine the proposed algorithm in larger networks operating under more hostile conditions.

Acknowledgments. This work is implemented within the framework of the Action "Supporting Postdoctoral Researchers" of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: a survey. Computer Networks 38(4), 393–422 (2002)
2. Karlof, C., Wagner, D.: Secure routing in wireless sensor networks: Attacks and countermeasures. In: First IEEE International Workshop on Sensor Network Protocols and Applications, pp. 113–127 (2002)
3. Camtepe, S.A., Yener, B.: Key distribution mechanisms for wireless sensor networks: a survey. Rensselaer Polytechnic Institute, Tech. Rep. (2005)
4. Shi, E., Perrig, A.: Designing secure sensor networks. IEEE Wireless Communications 11(6), 38–43 (2004)
5. Lazos, L., Poovendran, R.: Serloc: Robust localization for wireless sensor networks. ACM Transactions on Sensor Networks (TOSN), 73–100 (2005)
6. Dimitriou, T., Krontiris, I.: Secure In-network Processing in Sensor Networks. In: Security in Sensor Networks, pp. 275–290. CRC Press (August 2006)
7. Farooqi, A.H., Khan, F.A.: Intrusion detection systems for wireless sensor networks: A survey. In: Ślęzak, D., Kim, T.-H., Chang, A.C.-C., Vasilakos, T., Li, M., Sakurai, K. (eds.) FGCN/ACN 2009. CCIS, vol. 56, pp. 234–241. Springer, Heidelberg (2009)
8. Scarfone, K.A., Mell, P.M.: Sp 800-94. Guide to intrusion detection and prevention systems (idps). National Institute of Standards & Technology, Gaithersburg, MD, United States, Tech. Rep. (2007)
9. Rajasegarar, S., Leckie, C., Palaniswami, M.: Anomaly detection in wireless sensor networks. IEEE Wireless Communications 15(4), 34–40 (2008)
10. Hu, J.: Host-based anomaly intrusion detection. In: Handbook of Information and Communication Security, pp. 235–255 (2010)
11. Xie, M., Han, S., Tian, B., Parvin, S.: Anomaly detection in wireless sensor networks: A survey. J. Netw. Comput. Appl. 34(4), 1302–1325 (2011)
12. IEEE 802.15.4™ -2011: IEEE Stadard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) (2011)
13. Koubaa, A., Alves, M., Nefzi, B., Song, Y.-Q.: Improving the IEEE 802.15.4 Slotted CSMA/CA MAC for Time-Critical Events in Wireless Sensor Networks. In: Proceedings of the Workshop of Real-Time Networks (RTN 2006), Dresden, Germany (July 2006)
14. Sokullu, R., Korkmaz, I., Dagdeviren, O., Mitseva, A., Prasad, N.: An Investigation on IEEE 802.15.4 MAC Layer Attacks. In: Proceedings of the 10th International Symposium on Wireless Personal Multimedia Communications (WPMC 2007), pp. 1019–1023 (2007)
15. Jung, S.S., Valero, M., Bourgeois, A., Beyah, R.: Attacking beacon-enabled 802.15.4 networks. In: Jajodia, S., Zhou, J. (eds.) SecureComm 2010. LNICST, vol. 50, pp. 253–271. Springer, Heidelberg (2010)
16. Radosavac, S., Baras, J.S., Koutsopoulos, I.: A framework for MAC protocol misbehavior detection in wireless networks. In: Workshop on Wireless Security, pp. 33–42 (2005)
17. Sastry, N., Wagner, D.: Security considerations for IEEE 802.15.4 networks. In: Proceedings of the 3rd ACM Workshop on Wireless Security, WiSe 2004, pp. 32–42. ACM, New York (2004)
18. Alim, M.A., Sarikaya, B.: EAP-Sens: a security architecture for wireless sensor networks. In: Proceedings of the 4th Annual International Conference on Wireless Internet, WICON, pp. 1–9 (2008)

19. Sokullu, R., Dagdeviren, O., Korkmaz, I.: On the IEEE 802.15.4 MAC Layer Attacks: GTS Attack. In: Second International Conference on Sensor Technologies and Applications, SENSORCOMM 2008, pp. 673–678 (2008)
20. Amini, F., Misic, J., Pourreza, H.: Detection of sybil attack in beacon enabled IEEE 802.15.4 networks. In: International Wireless Communications and Mobile Computing Conference, IWCMC 2008, pp. 1058–1063 (2008)
21. da Silva, A.P.R., et al.: Decentralized intrusion detection in wireless sensor networks. In: Proceedings of 1st ACM International Workshop on Quality of Service and Security in Wireless and Mobile Networks (Q2SWINET 2005), pp. 16–23. ACM Press (2005)
22. Onat, I., Miri, A.: An intrusion detection system for wireless sensor networks. In: IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2005), vol. 3, pp. 253–259 (August 2005)
23. Yu, B., Xiao, B.: Detecting selective forwarding attacks in wireless sensor networks. In: Proceedings of the 20th International Conference on Parallel and Distributed Processing, IPDPS 2006, p. 351. IEEE Computer Society (2006)
24. Douceur, J.R.: The Sybil Attack. In: Druschel, P., Kaashoek, M.F., Rowstron, A. (eds.) IPTPS 2002. LNCS, vol. 2429, pp. 251–260. Springer, Heidelberg (2002)
25. Varga, A., Hornig, R.: An overview of the omnet++ simulation environment. In: Simutools 2008: Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops, pp. 1–10, ICST (2008)
26. Chen, F., Dressler, F.: A Simulation Model of IEEE 802.15.4 in OMNeT++. In: 6. GI/ITG KuVS Fachgesprach Drahtlose Sensornetze, Poster Session, pp. 35–38 (2007)
27. Hurtado-López, J., Casilari, E., Ariza, A.: Enabling IEEE 802.15.4 Cluster-Tree Topologies in OMNeT++. ICST, Brussels (2009)
28. Chipcon AS SmartRF© CC2420, preliminary datasheet, rev. 1.2 (2004), `http://www.chipcon.com/files/cc2420_data_sheet_1_3.pdf`