Energy-efficient Protection of IoT Networks using Trust Management on the IEEE 802.15.4 Protocol

Zeeshan Ali Khan
Department of Computer Science
National University of Computer and Emerging Sciences
Lahore, Pakistan
zeeshanali.khan@nu.edu.pk

Peter Herrmann
Information Security and Communication Technology
Norwegian University of Science and Technology (NTNU)
Trondheim, Norway
herrmann@ntnu.no

***Abstract*—Many devices used in the Internet of Things (IoT) have scarce processing resources and restricted energy budgets. Thus, established security mechanisms to protect IoT nodes against malicious attacks cannot be applied. Trust management is a promising technology to circumvent the processing limitations since it makes the use of lightweight but still powerful security mechanisms possible. In earlier work, we proposed a trust-based routing solution that builds the reputation of IoT devices to detect maliciously behaving ones. It has, however, the disadvantage of additional battery draining since it works only with constant idle listening of the communication channel which is quite energy intensive. In this paper, we present a method to alleviate this problem by aligning the proposed security algorithm with the popular IEEE 802.15.4 protocol that offers functionality to reduce active channel listening. In particular, we suggest an adaption to one of the application modes of IEEE 802.15.4 such that we can use our trust-based algorithm with often only slight losses from idle listening. The results of the protocol adaptations are discussed for two different scenarios.**

***Index Terms*—Internet of Things (IoT), Trust Management, Energy Efficiency, IEEE 802.15.4 Protocol**

I. Introduction

The *Internet of Things* (IoT) is an emergent technology in computing which is circumstantiated by the fast growing connection numbers of IoT devices [1]. However, like all distributed computing technology, IoT devices and applications are exposed to malicious attacks from intruding nodes. For instance, [2] describes a distributed Denial of Service (DDoS) attack that was successfully realized on Internet services using compromised IoT devices. Such attacks can be successfully carried out since most of the devices are small and have limited processing resources, energy budget, and bandwidth [3]. Hence, most of the security measures that are designed for traditional computer networks, cannot be used for them.

To detect the presence of maliciously behaving IoT nodes in an energy efficient and processing friendly manner, we proposed the use of an Intrusion Detection System (IDS) based on trust management [4], [5]. Our technique is devoted to the Routing Protocol for Low power and Lossy networks (RPL) [6]. It tackles selective forwarding and sinkhole attacks [7], [8], version number assaults [9], resp. self promoting, ballot stuffing, and bad mouthing attacks [10] that all can disrupt the IoT network traffic. To detect malicious behavior of a neighboring node, an IoT node constantly monitors the network traffic and checks if packets are forwarded timely and correctly. Based on the monitored behavior, the trust of a node in a neighbor increases or deteriorates. These trust relations are modeled as so-called trust values in form of opinion triangles in the Subjective Logic [11]. This formalism also allows us to aggregate trust values of different nodes in an entity. In this way, the general reputation of an entity can be described. If the reputation falls below a certain pre-defined threshold, the entity is considered as potentially malicious and will be isolated from the network.

978-1-6654-4399-9/21/$31.00 © 2021 IEEE

Initial analysis of the proposed approach suggested that it does not need sophisticated processing resources [4], [5]. Nevertheless, to build reputation of every IoT node present in the network, the devices need to dedicate their resources for active channel listening. Idle listening is a general problem of tiny devices with data reception capability since it demands a significant amount of energy [12]. Our approach aggravates this problem since it demands not only to listen for packets dedicated to a node itself but also to monitor packets that are directed to other nodes. Tests showed that the batteries of small devices deplete fairly quickly since idle listening consumes a lot of energy (see [13], [14]).

A way to reduce idle listening in general is to use specialized channel listening mechanisms in the Media Access Control (MAC) functionality of a protocol stack that minimize the active listening time of the devices. The most widely used MAC protocol for Wireless Sensor Networks and IoT Networks comprising tiny devices is IEEE 802.15.4 [15] which incorporates functionality to reduce the idle listening time effectively. Its most energy-preserving mode, however, cannot be used together with our trust-based security algorithm since it guarantees only that a station listens actively in time intervals in which packets are transmitted just for itself. Thus, a node cannot observe its neighbors and, in consequence, build trust or distrust in them. Fortunately, however, one can adapt IEEE 802.15.4 such that it can be combined with our trust-management approach also in its most energy friendly mode. This adaptation is the central contribution of this article.

We sketch the MAC protocol including its various operational modes in Sect. II. In Sect. III, we present the trust value computation using the Subjective Logic. Section IV

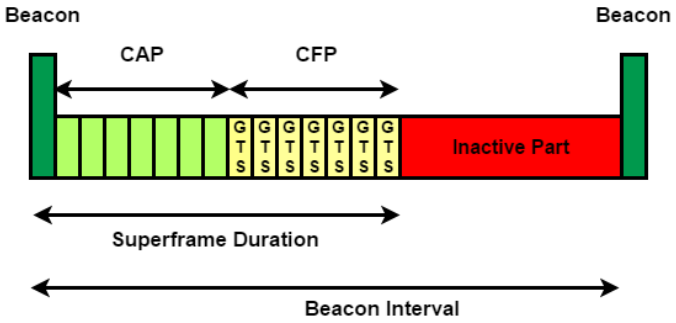

Fig. 1. An Example of superframe Structure for Beacon Enabled mode in IEEE 802.15.4.

describes the changes necessary to the IEEE 802.15.4 protocol. In Sect. V, we discuss how the combined mechanism effects the percentage of active channel listening times and, in consequence, the battery depletion times. The paper is completed with a look on related work and a conclusion.

II. IEEE 802.15.4 MAC Protocol

IEEE 802.15.4 [15] is used for so-called Low Rate Wireless Personal Area Networks (LR-WPAN) linking tiny IoT devices. It may be flexibly operated in several data transmission modes supporting a multiple range of embedded wireless sensing and control applications. Moreover, IEEE 802.15.4 allows for real-time data delivery. Further, it offers methods to reduce idle listening making the protocol energy friendly which is the most relevant property in our context.

The essential task of any MAC protocol is to control the mutual access of various nodes to the network medium. IEEE 802.15.4 uses the channel access algorithm Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). It also offers Guaranteed Time Slots (GTS) for real-time applications. With respect to the network nodes, one can distinguish between Full-Function Devices (FFD) and Reduced-Function Devices (RFD). The FFDs are expected to have more processing and energy resources than RFDs. Hence, they can carry out the more computing intensive parts of the protocol. The small and often resource-constrained IoT devices are typical RFDs that can only be involved in forwarding a reduced amount of data traffic. The protocol supports two types of topologies, i.e., star and peer-to-peer. When using a star topology, a network is normally managed by a so-called *Personal Area Network (PAN) coordinator* that has to be an FFD.

The IEEE 802.15.4 protocol offers three major modes. Two of them use special *beacon* frames to synchronize the network nodes. The beacons are sent by the PAN coordinator in fixed *Beacon Intervals* (BI). In this way, the coordinator is synchronized with the other nodes of the network. In order to decrease the channel listening time, the coordinator may also announce a portion of the BI as inactive. In this period, no communication takes place and, in consequence, the nodes may stop listening which reduces their energy consumption.

The nodes of a network are synchronized by a so-called *superframe* that is sent between two beacons, see Fig. 1. The superframe can be subdivided into a *Contention Access Period* (CAP) and a *Contention Free Period* (CFP). The data transmission in CAP follows the slotted CSMA/CA mechanism.

In the *Beacon Enabled Mode with GTS*, so-called *Guaranteed Time Slots* (GTS) are applied in the CFP. A GTS is normally reserved for applications requiring specific data bandwidth. It is the responsibility of the coordinator to allocate up to two slots per node during the CFP, one for transmitting and the other for receiving frames. Thus, since there are 16 GTSs in one superframe, up to eight nodes may be served in a BI. This scheme uses a polling mechanism that is controlled by the PAN coordinator. In this way, data packets with hard real-time deadlines may be transmitted in the CFP. The use of GTSs can reduce the idle listening time of the end points significantly since a node needs only to listen to the CAP and to the GTSs assigned to itself but may sleep otherwise. The PAN coordinator, however, has to stay in active mode as it is responsible for beacon and time slot management for the member nodes.

The *Beacon Enabled Mode without GTS* does not use the CFP and GTSs. Thus, all data transfer is carried out in the CAP. Since a station has to listen to the full CAP, i.e., to the whole data transfer, the energy reduction here is expected to be lower than when using GTSs.

IEEE 802.15.4 may also be operated in the *Non-beacon Enabled Mode*. Here, no beacon frames are transmitted, and the nodes send data to the PAN coordinator using the unslotted CSMA/CA mechanism. The coordinator is only responsible for device association and disassociation. While the Non-beacon Enabled Mode permits scalability and self organization, it cannot ensure real-time data frame delivery. Moreover, the node needs to constantly listen to the channel in order to receive packets directed to it such that there will be no significant energy preservation.

III. Trust Value Computation for IoT Networks

Trust values can be used to represent the degree of trust of an entity in another one on computers. The trust values can be discrete like in eBay, where the reliability of sellers and buyers is described by a number of different stars. Alternatively, they can be continuous which allows for a more fine-grained description of a trust relation.

For our algorithm, we use the opinion triangles of the Subjective Logic [11] that makes it possible to describe the trust and distrust in an entity but also to consider the uncertainty of a trust verdict. An *opinion triangle* (see Fig. 2) consists of three variables b, d, and u. The variable b describes the quantity of *belief (trust)*, while d refers to the *disbelief (distrust)*, and u expresses the *uncertainty* of the trust value. All three variables are real numbers within the interval between 0 and 1 and their sum must always be equal to 1. Hence, a trust value can be represented as a dot within a unilateral triangle.

Trust that is based on direct experience with an entity is called direct trust. It can be computed from experience values

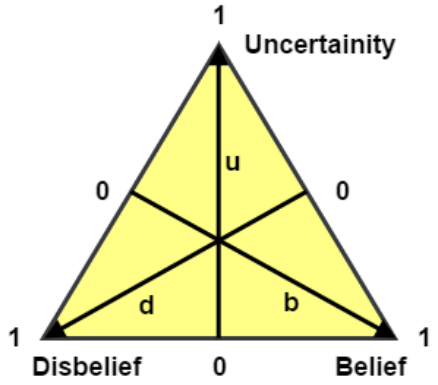

Fig. 2. Opinion triangle in the Subjective Logic [11].

using metrics like the following one [16]:

$$b = \frac{p}{p+n+k} \qquad d = \frac{n}{p+n+k} \qquad u = \frac{k}{p+n+k}$$

Here, the number p of positive and n of negative interactions with an entity are counted and the variables of an opinion triangle are computed from them. The speed of overcoming uncertainty due to a limited number of direct experience reports can be defined using the constant k for which often the values 1 or 2 are selected. The metric can also incorporate a forgetting factor (see, e.g., [17]) such that older experiences are eliminated, or it may remember the whole event history.

We use the Subjective Logic to build the reputation of the PAN coordinators in the IEEE 802.15.4 protocol. In particular, we consider selective forwarding attacks [7], [8], in which a PAN coordinator fails to operate correctly and drops the packets that are destined to other nodes. The used trust building mechanism is described by Algorithm 1. When sending a packet to the PAN coordinator for forwarding, a node keeps the packet identifier and recipient information in a buffer called $send\_stack$. Thereafter, the node listens to the data traffic generated by the PAN coordinator to find out if the packet is correctly and timely forwarded by the coordinator. If that is the case, the packet information is deleted from $send\_stack$ and the variable p for the trust value of the PAN coordinator is incremented since our node had a positive experience with it. If the packet, however, is not forwarded within a certain time limit or falsified in a certain way, the variable n is incremented indicating a negative experience. Further, the packet information is removed from the $send\_stack$ as well. From the variables p and n, an opinion triangle showing the trust of our node into the PAN controller is computed using the metric introduced above.

Based on the interaction of the associated devices with a PAN coordinator, each one gets a distinct trust value for this device. The nodes forward their opinion triangles periodically to an aggregator node, preferably an FFD, which combines these values using Algorithm 2. This node aggregates the received opinion triangles about a certain PAN coordinator using the consensus operator $\oplus$ of the Subjective Logic [11]:

Algorithm 1 Trust Computing in a Member Node.

```
while True do
  if packet_sent then
    Store packet in send_stack
  end if
  if packet_received then
    if packet_transmitter ∈ PAN_coordinator then
      if packet ∈ send_stack then
        Remove packet from send_stack
        Increment value p of PAN_coordinator
      end if
    end if
  end if
  if Timeout of a packet in send_stack then
    Remove packet from send_stack
    Increment value n of PAN_coordinator
  end if
end while
```

Algorithm 2 Trust Computing in the Aggregator Node.

```
if Trustvalues packets are received from member nodes then
  Combine trust values for every member node
  for All PAN coordinators do
    if Disbelief > intruder_threshold then
      Notify operator
    end if
  end for
end if
```

If $v_1 = (b_1, d_1, u_1)$ is the trust value of a node x in a node y, and $v_2 = (b_2, d_2, u_2)$ the trust value of a node w in the same node y, the combined trust of x and w in y can be expressed by $v_1 \oplus v_2$ as follows:

$$\left(\frac{b_1 u_2 + b_2 u_1}{u_1 + u_2 - u_1 u_2}, \frac{d_1 u_2 + d_2 u_1}{u_1 + u_2 - u_1 u_2}, \frac{u_1 u_2}{u_1 + u_2 - u_1 u_2} \right)$$

This operator is associative and commutative. Hence, it can be used to aggregate all trust values concerning node y. In this way, the general reputation of y based on the input of various other entities can be computed. If the disbelief or distrust d of a PAN coordinator exceeds a certain threshold, the aggregator disseminates this information to the network operator which can then neutralize the malicious or malfunctioning coordinator.

IV. Adapting the IEEE 802.15.4 Protocol

With respect to energy consumption, the most relevant issue of our approach is the fact that Algorithm 1 requests a device to listen to the channel also after transmitting a message. Only then it can find out whether the PAN coordinator, indeed, forwards this message correctly and within the desired time limit. In the following, we will sketch how this demand impacts the energy consumption in the three modes of the IEEE 802.15.4 protocol introduced in Sect. II. In particular,

we distinguish between nodes that both, transmit and receive, and devices that only send data, e.g., typical IoT sensors.

In the Non-beacon Enabled Mode, all nodes that receive data have to listen constantly to the channel to catch all packets directed to them. Thus, there is no difference when also certain other packets are collected and checked according to Algorithm 1. Thus, we can use this mode without changing the listening policy for our approach. Nodes that just transmit but do not receive messages, however, usually listen only into the active channel when they intend to send messages but fall asleep otherwise. This does not support Algorithm 1 since such a node would likely sleep when its frames are forwarded. One can, however, change this mode by letting the node actively listen on the channel after sending a message until it either receives the message forwarded by the coordinator or gets a timeout indicating that the message was not timely forwarded. Of course, the longer listening times entail increased battery draining.

In the Beacon-Enabled Mode without GTS, the period between two beacons are segmented into an active and a passive phase, and nodes are only listening to the channel in the active phase which leads to a general energy reduction. That holds both, for nodes that send and receive and those that only send. In the active phase, the functionality is identical to the Non-beacon Enabled Mode since stations that receive packets have to listen continuously in the active part. Devices that only send packets, will have to be active until their message is forwarded or the timeout occurred.

In the Beacon-Enabled Mode with GTS, a node that receives packets, is active during the CAP part of the superframe in which it learns if slots are dedicated to itself. Further, it has to be active during the slots assigned to itself but can sleep during other slots as well as in the passive phase after finishing the superframe. The pristine use of this idle listening strategy does not support Algorithm 1 since the forwarding of the massages from the coordinator to the recipients is missed. We can, however, easily adapt this strategy to our trust-based approach by extending the active listening to the reception channels of all stations for which open messages are stored in the variable $send\_stack$. Using this extension to the idle listening procedure, all forwarded messages are, indeed, correctly detected.

Nodes that only send packets, sleep continuously as long as they don't want to transmit. When there is a packet to be transmitted, the node gets active and calls the PAN coordinator by a message in the CAP in order to receive a pair of slots in the CFP through which the message can be delivered. After finishing the transmissions, the PAN coordinator is notified to remove the slots. Using this mode without our security approach, the node can stop listening directly after the message about removing the slot. When the device shall support Algorithm 1, it has to stay awake and listen to the slots for the receiver of the message until the packet is either sent or the timeout occurs.

As long as the messages transmitted by non-receiving nodes can be transmitted in a single packet, the Beacon-Enabled Mode with GTS seems to be too cumbrous due to the need to negotiate the assignment of slots. With respect to energy friendliness, however, this mode seems to be best for the combination with our trust manage-based approach independently of the station type. The reason is that, except for reading the beacons and the CAP of a superframe, a node has to listen only during slots that contain relevant information for itself, i.e., its own slots as well as the receiving slots of the recipients of its messages. We expect that the Beacon Enabled Mode without GTS scores second since a node has to listen to communication with uninvolved stations but, at least, it can sleep in the passive period. The Non-beacon Enabled Mode demands continuous reading such that we do not expect advantages in comparison to our original RPL-based work [4] here. In the following, we will discuss the results of the analysis of the three modes to find out if our expectations are correct.

V. Analysis

With help of MATLAB, we simulated IEEE 802.15.4 with a star topology consisting of eight nodes connected to a PAN coordinator. To simulate a selective-forwarding attack, we made the coordinator malicious and let it only forward 50% of the traffic in average. By using the metric of [16] introduced in Sect. III, the reputation for the coordinator should approximate the trust value $(0.5, 0.5, 0)$. Depending on the selected threshold for the disbelief value d, the coordinator will declare the coordinator as malicious and notify the network operator long before coming close to this trust value. We did not investigate in suitable policies, yet, but common sense tells us that d should not be larger than 0.2 since in that case more than every fifth message is not correctly forwarded. Here, however, one might further demand that the uncertainty value u has to be below a certain value in order to prevent false accusations of a coordinator due to some negative experiences in the very beginning. If we assume that the constant k is set to 2 in the metric, 0.1 could be a good threshold for u since then alarms are only raised if there have at least been 18 interactions. In this case, we should be sufficiently certain about the trustworthiness of the PAN controller.

In our experiments described in the following, we assume that 50% of the BI is passive in the two modes using beacons. Further, we expect that in the Beacon Enabled Mode with GTS, the CFP with the slots comprises 80% of the active phase and the beacon resp. CAP the remaining 20%. Finally, the timeout interval within which a packet has to be forwarded to prevent a negative evaluation is set to eight seconds.

Based on these numbers, we conducted two series of experiments to test energy usage for the alignment of our trust-based approach with the three modes of IEEE 802.15.4:

1) This experiment considers nodes that constantly transmit and receive data as one can find for instance in ad-hoc networks (see [18]). We suppose that each node sends a packet every two seconds arbitrarily to one of the other stations.
2) In this case, we analyze nodes that just transmit data but do not receive packets like that is the case for many IoT

TABLE I
AWAKE TIME PERCENTAGES FOR THE DIFFERENT IEEE 802.15.4 MODES.

Scenario	Mode	Not supporting our approach	Supporting our approach
Read and write	Non-beacon Enabled Mode	100.0%	100.0%
	Beacon Enabled Mode without GTS	50.8%	50.8%
	Beacon Enabled Mode with GTS	15.3%	19.6%
Read only	Non-beacon Enabled Mode	0.8%	7.6%
	Beacon Enabled Mode without GTS	0.9%	4.4%
	Beacon Enabled Mode with GTS	0.1%	1.1%

sensors. Here, we assume that a node wakes up once a minute to send a packet to a certain station. The overall time to get access to the channel and to transmit the packet is estimated as 500 milliseconds.

In the following subsections, we describe the results of our analysis based on these numbers. First, the analyzed idle listening percentages of the nodes are introduced. Thereafter, we discuss the impact of idle listening for the lifetimes of the batteries.

A. Listening Time Percentages

We carried out simulations with MATLAB for both scenarios and all three IEEE 802.15.4 modes which are depicted in Table I. For a better comparison, we list in the third column of the table the percentaged awake times, when IEEE 802.15.4 is used without our IDS approach. In the forth column, we depict the awake time percentages using the amended idle listening procedures to support Algorithm 1 (see Sect. IV).

Not surprisingly, with respect to the experiments for the first scenario, in which a node sends and receives messages, in the Non-beacon Enabled Mode the node listens the whole time independently of supporting our security approach or not. The same holds for the listening time percentages when using the Beacon Enabled Mode without GTS. Here, the values are a little larger than the expected 50% since a node has to start active channel listening earlier in order to prevent missing the next beacon. A significant difference between the plain transmission and the support of our approach can be found in the Beacon Enabled Mode with GTS. The reason for this is evident: In the plain mode, the node has to listen to the CAP and only the two slots of the CFP dedicated to itself. In contrast, in the solution assisting Algorithm 1, also the slots towards a station for which a packet is stored in variable $send\_stack$ have to be monitored. If only every second packet is forwarded on time by the coordinator, a node has in average packets of 1.4 recipient nodes in its $send\_stack$ (see Algorithm 1). Therefore, it must listen to 3.4 nodes in average which explains the difference. Nevertheless, this mode is far more energy friendly than the other two.

A similar effect can also be seen for the second scenario in which a node only transmits messages every minute. Of course, the percentages are generally lower since the node sleeps most of the time. Unlike the first scenario, however, the support of the security approach increases the listening times also in the Non-beacon Enabled Mode and the Beacon Enabled Mode without GTS. The reason for this is that, whenever the PAN coordinator does not forward a packet, the node needs to keep continuing listening the eight seconds until the timeout occurs. This effect is also mainly responsible for the increase of the listening time percentage in the Beacon Enabled Mode with GTS. Here, the growth is much more distinct than in the first scenario.

Altogether, the expected order, i.e., the Beacon Enabled Mode with GTS scores first, the Beacon Enabled Mode without GTS second, and the Non-beacon Enabled Mode third with respect to idle time percentages, holds for both scenarios. In particular, the Beacon Enabled Mode with GTS gives values that are about 80% better than the Non-beacon Enabled Mode and our previous RPL-based solution [4], [5].

B. Battery Lifetime Estimation

To get a more in-depth sense of the practical effects of our adaptation to IEEE 802.15.4, we computed the effects of the three modes on the battery lifetime for both scenarios. We assume devices communicating via CC2480 transceivers and used the procedure from [19] for our computations. With respect to the battery consumption for communication, one can distinguish the three states *idle listening*, *transmission of packets*, and *sleeping*. (In principle, one could also consider the switching between sleeping and listening as a state but since these switches are very short and do not need a lot of energy, we do not consider them in our computation.) The current $I_{listening}$ for idle listening is 32.5 mA, while $I_{transmission}$ for packet transmission is 30.5 mA, and I_{sleep} when the node sleeps, corresponds to 0.00075 mA. Since the time of sending packets is short in comparison with the listening time and the two currents are relatively close, we simplified our computations and distinguish only between sleeping and being active. Thus, we assume for the current I_{active} when the node is active: $I_{active} \approx I_{listening} = 32.5$ mA. We can define the average battery drain I_{drain} as follows:

$$I_{drain} = \frac{t_{active} I_{active} + t_{sleep} I_{sleep}}{t_{active} + t_{sleep}}$$

Analogous to the currents, t_{active} refers to the time, the node is listening (or transmitting) actively while t_{sleep} refers to the time, the node sleeps.

TABLE II
LIFETIME OF A $10Ah$ BATTERY IN THE DIFFERENT IEEE 802.15.4 MODES.

Scenario	Mode	Not supporting our approach	Supporting our approach
Read and write	Non-beacon Enabled Mode	307.7 h	307.7 h
	Beacon Enabled Mode without GTS	605.7 h	605.7 h
	Beacon Enabled Mode with GTS	1935.9 h	1569.7 h
Read only	Non-beacon Enabled Mode	36243.7 h	3949.1 h
	Beacon Enabled Mode without GTS	26322.2 h	6985.3 h
	Beacon Enabled Mode with GTS	228365.0 h	28740.9 h

If we neglect the current required in the start up phase, the lifetime of a battery having capacity C can be approximated for the different IEEE 802.15.4 modes as follows:

$$L = \frac{C}{I_{drain}}$$

A typical capacity of reloadable batteries used, e.g., in modelling, is 20 Ah. Thus, it is realistic to assume that 10 Ah are available just for a CC2480 transceiver in a node, and we use this value as the battery capacity C in our computations. We can now easily calculate the average drain for the three modes in IEEE 802.15.4 from the values in Table I using the two formulas depicted above.

The results based on the simulated mean average idle times are listed in Table II. In the scenario in which the node both, transmits and receives data, the listening time adaptations presented in Sect. IV seem acceptable. While this alignment leads to a loss of around 20% in the Beacon Enabled Mode with GTS, the energy preservation capabilities of this mode results still in a lifetime that is five times better than in the Non-beacon Enabled Mode (65 instead of 13 days).

For the experiments with an IoT node that only transmits, however, the results are very different. Here, adapting the three modes to Algorithm 1 reduces the battery lifespan by up to nearly 90% which is hardly acceptable. These losses even overbalance the gain through using the Beacon Enabled Mode with GTS instead of the Non-beacon Enabled Mode since that gives a reduction of around 20% (three instead of four years).

A reason for these disillusioning results seems to be, however, that the read-only scenario is unrealistically harsh. Since the cause for using our security approach is exactly to find out maliciously behaving PAN coordinators, one should expect that such a coordinator will be quickly replaced. Thus, the nodes will have the, in average, four seconds long waiting times after transmitting their data only for a relatively short time and the effective battery lifetimes should be drastically better.

Moreover, our algorithm creates a new vulnerability. If attackers manage to guess the timeout interval correctly, they can set a PAN coordinator to forward frames with voluntary delays that are just a little shorter than this interval, e.g., 7.8 seconds in our example. Then the idle listening times of the nodes would still be significantly prolonged without deteriorating the reputation of the coordinator. While this attack can be mitigated by randomly varying timeout intervals, one should be aware of potential impacts of the adaptations to tiny sensor nodes and contemplate if it is useful to use such nodes for building reputations of PAN coordinators. That holds particularly, if the sensors are at hardly accessible locations such that changing batteries is difficult.

VI. RELATED WORK

While the field of intrusion detection for IoT networks is still quite novel, already some approaches were published in the recent years. Similar to our previous work [4], [5], Cervantes et al. use trust management and the building of reputations for intrusion detection [20]. They also developed a solution targeting sinkhole attacks on the routing layer of RPL. In contrast to us, they build the reputation of nodes depending on retransmission rates and not on observing the behavior of other nodes. While this approach prevents the idle listening problem, it seems to be less precise about the reason for errors. Further, it allows the detection of fewer attack types, e.g., no version number attacks (see [9]). Instead of the Subjective Logic, they compute the reputation of nodes using the $Beta(\alpha, \beta)$ distribution.

Other IDSs for IoT networks are based on distributed technology, statistical detection, resp. game theory. A distributed and cooperative IDS is introduced in [21]. It protects IoT networks applying artificial immunity mechanisms in form of attack libraries to which the sensed behavior is compared. The authors of [22] provide a similar approach which, however, is based on penetration testing while [23] use network graph inconsistency detection to target well-known routing attacks like sinkhole, selective forwarding, and spoofing.

An IDS based on statistical methods to detect behavioral anomalies in IoT-driven smart homes is presented in [24]. In particular, behavioral models are created using immunity-inspired algorithms that then are compared with the data actually sensed. [25] presents a distributed IDS that analyzes anomaly data to detect attacks on the perception layer of a network. Similarly, the authors of [26] propose a lightweight anomaly mining algorithm which seems to be very friendly to the computing resources of tiny nodes. Another approach based on anomalies is introduced in [27]. This IDS detects abnormal packets by matching bit patterns using a lookup table.

The approach introduced in [28] uses game theory to predict whether a new attack will occur. Since, the energy-intensive anomaly detection is only activated when an assault is likely

to come, this approach minimizes the overall energy consumption. In contrast to our approach, however, this reduction may lead to missed attacks. Another IDS models attacks of varying seriousness as a Bayesian game [29]. The results of this game can then be used to determine the gravity of an attack such that an adequate course of action can be selected.

VII. Conclusion

In this article, we discussed how the impact of our trust management approach for detecting malicious PAN coordinators on battery consumption can be mitigated using the MAC protocol IEEE 802.15.4 with an adapted idle listening strategy. The results of our analyses show that the Beacon Enabled Mode with GTS of this protocol is very helpful, in particular, for nodes that not only transmit data but also regularly receive packets. The alignment of the approach with IEEE 802.15.4 has a good potential to reduce the energy consumption of idle listening in spite of the necessity to consider a slightly higher number of frames.

In this paper, we based our conclusions on analysis and simulation. The next step is, of course, to validate the approach with real systems. We currently build a test-bed that consists of Z1 devices running on the operating system Contiki. With that, it should not be difficult to measure the energy drain of the batteries in the different modes of the IEEE 802.15.4 protocol. That will allow us to find out whether the predictions given here, indeed, hold in practice.

References

[1] T. Rebbeck, M. Mackenzie, and A. Ali, "Predictions for IoT: Investments in NB-IoT, LTE-M and New Capabilities Prepare Operators for an Active 2018," http://www.analysysmason.com/Research/Content/Comments/Predictions-2018-IoT-RDME0-RMA17/, 2017, accessed: 2018-05-11.

[2] A. Nordrum, "What Is a Distributed Denial-of-Service Attack and How Did It Break Twitter?" IEEE Spectrum, https://spectrum.ieee.org/tech-talk/telecom/security/what-is-a-distributed-denialofservice-attack-and-how-did-it-break-twitter, 2016, accessed: 2018-05-13.

[3] S. Raza, H. Shafagh, K. Hewage, R. Hummen, and T. Voigt, "Lithe: Lightweight Secure CoAP for the Internet of Things," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3711–3720, 2013.

[4] Z. A. Khan and P. Herrmann, "A Trust Based Distributed Intrusion Detection Mechanism for Internet of Things," in *31st IEEE International Conference on Advanced Information Networking and Applications (AINA)*. Taipei: IEEE Computer, March 2017, pp. 1169–1176.

[5] Z. A. Khan, J. Ullrich, A. G. Voyiatzis, and P. Herrmann, "A Trust-based Resilient Routing Mechanism for the Internet of Things," in *International Conference on Availability, Reliability, and Security*, 2017, p. article 27.

[6] IETF, "RfC 6550 — RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," https://tools.ietf.org/html/rfc6550, 2012, accessed: 2016-10-24.

[7] C. Karlof and D. Wagner, "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures," *Ad Hoc networks*, vol. 1, no. 2, pp. 293–315, 2003.

[8] L. Wallgren, S. Raza, and T. Voigt, "Routing Attacks and Countermeasures in the RPL-based Internet of Things," *International Journal of Distributed Sensor Networks*, vol. 9, no. 8, p. 11 pages, 2013.

[9] A. Mayzaud, A. Sehgal, R. Badonnel, I. Chrisment, and J. Schönwälder, "A Study of RPL DODAG Version Attacks," in *IFIP International Conference on Autonomous Infrastructure, Management and Security*. Springer-Verlag, 2014, pp. 92–104.

[10] R. Chen, J. Guo, and F. Bao, "Trust Management for SOA-based IoT and its Application to Service Composition," *IEEE Transactions on Services Computing*, vol. 9, no. 3, pp. 482–495, 2016.

[11] A. Jøsang, "A Logic for Uncertain Probabilities," *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems*, vol. 9, pp. 279–311, 2001.

[12] K. R. Lai, P. K. Sahoo, C. Y. Chang, and C. C. Chen, "Reduced Idle Listening Based Medium Access Control Protocol for Wireless Sensor Networks," in *International Conference on Communications and Mobile Computing*, vol. 3, 2010, pp. 329–333.

[13] Z. A. Khan and U. Abbasi, "Evolution of Wireless Sensor Networks toward Internet of Things," in *Emerging Communication Technologies Based on Wireless Sensor Networks: Current Research and Future Applications*. CRC Press, 2016, ch. 7, pp. 179–200.

[14] Z. A. Khan and M. Auguin, "A multichannel design for qos aware energy efficient clustering and routing in wmsn," *Int. J. Sen. Netw.*, vol. 13, no. 3, pp. 145–161, Jun. 2013.

[15] J. Bhar, "A Mac Protocol Implementation for Wireless Sensor Network," *Journal of Computer Networks and Communications*, vol. 2015, no. 1, 2015.

[16] A. Jøsang and S. J. Knapskog, "A Metric for Trusted Systems," in *21st National Security Conference*. NSA, 1998.

[17] P. Herrmann, "Temporal Logic-Based Specification and Verification of Trust Models," in *4th International Conference on Trust Management (iTrust)*, ser. LNCS 3986. Springer–Verlag, 2006, pp. 105—119.

[18] P. Papadimitratos and Z. J. Haas, "Secure Message Transmission in Mobile Ad hoc Networks," *Ad Hoc Networks*, vol. 1, pp. 193–209, 2003.

[19] E. Casilari, J. M. Cano-García, and G. Campos-Garrido, "Modeling of Current Consumption in 802.15.4/ZigBee Sensor Motes," *Sensors*, vol. 10, no. 6, pp. 5443–5468, 2010.

[20] C. Cervantes, D. Poplade, M. Nogueira, and A. Santos, "Detection of Sinkhole Attacks for Supporting Secure Routing on 6LoWPAN for Internet of Things," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE Computer, 2015, pp. 606–611.

[21] C. Liu, J. Yang, R. Chen, Y. Zhang, and J. Zeng, "Research on Immunity-based Intrusion Detection Technology for the Internet of Things," in *7th International Conference on Natural Computation (ICNC)*, vol. 1. IEEE Computer, 2011, pp. 212–216.

[22] P. Kasinathan, C. Pastrone, M. A. Spirito, and M. Vinkovits, "Denial-of-Service Detection in 6LoWPAN based Internet of Things," in *IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE Computer, 2013, pp. 600–607.

[23] S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Real-time Intrusion Detection in the Internet of Things," *Ad Hoc Networks*, vol. 11, no. 8, pp. 2661–2674, 2013.

[24] B. Arrington, L. Barnett, R. Rufus, and A. Esterline, "Behavioral Modeling Intrusion Detection System (BMIDS) Using Internet of Things (IoT) Behavior-Based Anomaly Detection via Immunity-Inspired Algorithms," in *25th International Conference on Computer Communication and Networks (ICCCN)*. IEEE Computer, 2016, pp. 1–6.

[25] R. Fu, K. Zheng, D. Zhang, and Y. Yang, "An Intrusion Detection Scheme based on Anomaly Mining in Internet of Things," in *4th IET International Conference on Wireless, Mobile & Multimedia Networks (ICWMMN)*. IET, 2011, pp. 315–320.

[26] Y. Liu and Q. Wu, "A Lightweight Anomaly Mining Algorithm in the Internet of Things," in *5th IEEE International Conference on Software Engineering and Service Science (ICSESS)*. IEEE Computer, 2014, pp. 1142–1145.

[27] D. H. Summerville, K. M. Zach, and Y. Chen, "Ultra-Lightweight Deep Packet Anomaly Detection for Internet of Things Devices," in *IEEE 34th International Performance Computing and Communications Conference (IPCCC)*. IEEE Computer, 2015, pp. 1–8.

[28] H. Sedjelmaci, S. M. Senouci, and M. Al-Bahri, "A Lightweight Anomaly Detection Technique for Low-resource IoT Devices: A Game-theoretic Methodology," in *IEEE International Conference on Communications (ICC)*. IEEE Computer, 2016, pp. 1–6.

[29] Q. D. La, T. Quek, J. Lee, S. Jin, and H. Zhu, "Deceptive Attack and Defense Game in Honeypot-enabled Networks for the Internet of Things," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1025–1035, 2016.