ZigBee IoT Intrusion Detection System: A Hybrid Approach with Rule-based and Machine Learning Anomaly Detection

Fal Sadikin and Sandeep Kumar
Signify Research, Eindhoven, The Netherlands

Keywords: ZigBee IoT Intrusion Detection System, Rule-based Method, Machine Learning Anomaly Detection.

Abstract: The Internet of Things (IoT) is an emerging technology with potential applications in different domains. However these IoT systems introduce new security risks and potentially open new attack vector never seen before. In this article, we show various methods to detect known attacks, as well as possible new types of attacks on ZigBee based IoT systems. To do so, we introduce a novel Intrusion Detection System (IDS) with hybrid approach by combining the human-crafted rule-based and machine learning-based anomaly detection. Rule-based approach is used to provide accurate detection mechanism for known attacks, but the rule-based approach introduces complexity in defining precise rules for accurate detection. Therefore, machine learning approach is specifically used to create a complex model of normal behaviour that is used for anomaly detection. This paper outlines the IDS implementation that cover various types of detection methods both to detect known attacks, as well as potential new type of attacks in the ZigBee IoT systems.

1 INTRODUCTION

The Internet of Things (IoT) is emerging as key enabler for unique solutions in many application domains like home and building automation. However, due to its device constraints and their massive deployments, the IoT also introduces new security issues that could open new attack vectors never seen before. This paper introduces a novel prototype of Intrusion Detection Techniques, which is specifically tailored to a ZigBee (ZigBee-Alliance, 2015) based IoT system.

Intrusion Detection System (IDS) is a well-known technology for detecting attacks in digital or connected systems. In general, there are two approaches of IDS in term of detection methods: misuse rule-based and anomaly-based Machine Learning (ML). Misuse rule-based are human created rules to detect both known and unknown attacks. While, anomaly-based machine learning IDS is a detection method based on pre-defined ML model of normal behaviour, which are used to detect malicious activities or new type of attacks that have never been seen before.

This paper outlines our prototype of a novel detection technique for a ZigBee based IoT system that combines misuse rule-based and ML anomaly detection to detect both known and unknown attacks. In this prototype, we use specific features of ZigBee protocol to create the rules for the rule-based detection and to train the ML model to detect the attacks. Specifically, the main contributions of this paper is four-fold:

- Survey various attacks and possible exploitation scenarios in a ZigBee based IoT system.
- Data-sets of network behavior when some of these attack scenarios are performed on the ZigBee based IoT system.
- Novel rule-based attack detection techniques tailored for a ZigBee based IoT system.
- Novel machine learning based anomaly detection techniques for a ZigBee based IoT system.

2 RELATED WORK

Our work is related to several research domains as ZigBee Intrusion Detection System is based on various technologies like wireless communication systems, data analytics, machine learning and the detection techniques itself. In this section, we briefly outline several existing solutions that relate to our work.

(Pacheco, 2016) introduces an Anomaly Behaviour Analysis (ABA) Intrusion Detection System (ABA-IDS) to detect anomalies in IoT system. This approach can detect known and unknown attacks for

Sadikin, F. and Kumar, S.
ZigBee IoT Intrusion Detection System: A Hybrid Approach with Rule-based and Machine Learning Anomaly Detection.
DOI: 10.5220/0009342200570068
In *Proceedings of the 5th International Conference on Internet of Things, Big Data and Security (IoTBDS 2020)*, pages 57-68
ISBN: 978-989-758-426-8

IoT end nodes, with high detection rate and low false alarms.

(Doohwan-Oh, 2014) presents a lightweight security system that uses a novel malicious pattern-matching engine. The authors manage to limit the memory usage of the proposed system in order to make it work on resource-constrained devices. To mitigate performance degradation due to limitations of computation power and memory, the authors propose two novel techniques, auxiliary shifting and early decision.

(T-H-Lee, 2014) proposes a lightweight intrusion detection model based on analysis of node's energy consumed in a 6LowPAN network. The 6LoWPAN energy consumption models for mesh-under and route-over routing schemes are created. The sensor nodes with irregular energy consumption are identified as malicious attackers.

(Summerville, 2015) have developed an ultra-lightweight deep packet anomaly detection approach that is feasible to run on resource constrained IoT devices, but still provides good discrimination between normal and abnormal payloads. Due to its simplicity, the approach can be efficiently implemented in either hardware or software and can be deployed in network appliances, interfaces, or in the protocol stack of a device.

(Pongle, 2015) propose a novel intrusion detection system for the IoT, which can detect a wormhole attack and the attacker. The proposed methods use the location information of node and neighbour information to identify the wormhole attack and received signal strength to identify attacker nodes.

(Anhtuan-Le, 2016) propose a specification to detect Routing Protocol for Low power and Lossy network (RPL) topology attacks that can downgrade the network performance significantly by disrupting the optimal routing structure.

(Rathore, 2018) introduce a fog-based attack detection framework that relies on the fog computing paradigm and a newly proposed ELM-based Semi-supervised Fuzzy C-Means (ESFCM) method. As an extension of cloud computing, fog computing enables attack detection at the network edge and supports distributed attack detection.

(Chawla, 2018) propose a platform intrusion detection system that uses machine learning algorithms to detect security anomalies in IoT networks. This detection platform provides security as a service and facilitates inter-operability between various network communication protocols used in IoT.

(A-A-Diro, 2018) propose design and implementation of deep learning based distributed attack detection mechanism, which reflects the underlying distribution features of IoT. Moreover, (Maniriho and Ahmad, 2018) also studies the Performance of Machine Learning Algorithms in Anomaly Network Intrusion Detection System. Furthermore, other research works propose various detection techniques including Probabilistic-driven Ensemble Approach proposed by (Saia et al., 2018), and IDS with Internet-integrated CoAP Sensing Applications proposed by (Granjal and Pedroso, 2018).

In general, there is a lot of research work that addresses various Intrusion Detection System approaches reaching from rule-based detection, anomaly-based detection, to machine learning and deep learning. However, to the best of our knowledge, none of them address specific techniques to detect attacks on large-scale ZigBee IoT system under its various constraints.

3 IoT ZigBee SECURITY

3.1 ZigBee Protocol

This section describes the ZigBee stack architecture (ZigBee-Alliance, 2015) and network topology defined in the ZigBee standard specification provided by ZigBee Alliance.

3.1.1 ZigBee Stack Architecture

As described in ZigBee Specification (ZigBee-Alliance, 2015), the ZigBee Alliance has developed a very low-cost, very low power consumption, wireless communications standard. Solutions adopting the ZigBee standard are embedded in consumer electronics, home and building automation, industrial controls, medical sensor applications, toys, and games.

The ZigBee stack architecture is defined based on a set of layers. Each layer performs a specific set of services for the layer above and below. Figure 1 represents the outline of the ZigBee Stack Architecture. Basically, ZigBee Stack Architecture is built based on two standards. The lower layers, which are the Physical Layer (PHY) and the Medium Access Control (MAC) are defined by IEEE 802.15.4 standard. The upper layers, which are Network Layer (NWK) and Application Layer (APL) are defined by ZigBee Alliance itself. Furthermore, ZigBee also offers network layer and application layer security.

ZigBee PHY uses three different frequency ranges. The lower frequency band 868 MHz is used in Europe and 915 MHz band is used in several countries such as United Stated and Australia. Furthermore, the higher frequency band in 2.4 GHz is used worldwide.

Figure 2 depicts the 2.4 GHz ZigBee channels, where the channels are divided into 16 different channels, ranging from channel 11 at 2405 MHz to channel 26 at 2480 MHz.

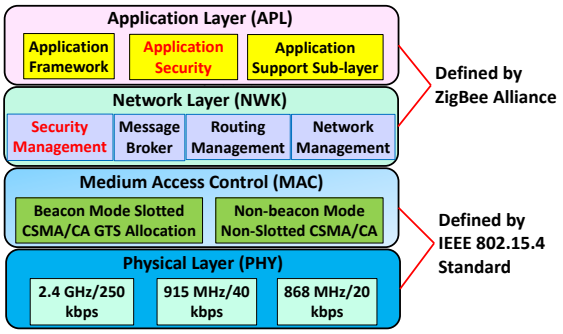

Figure 1: ZigBee Stack Architecture.

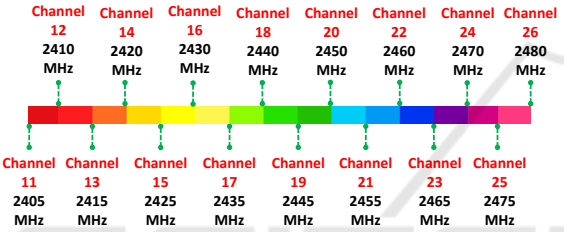

Figure 2: 2.4 GHz ZigBee Channels.

3.1.2 ZigBee Network Topology

ZigBee Specification (ZigBee-Alliance, 2015) also defines that ZigBee network layer (NWK) support star, tree, and mesh topologies. In a star topology, the network is controlled by one single device called the ZigBee coordinator. The ZigBee coordinator is responsible for initiating and maintaining the devices on the network. All other devices, known as end devices, directly communicate with the ZigBee coordinator. In mesh and tree topologies, the single ZigBee coordinator is responsible for starting the network and for choosing certain key network parameters, but the network may be extended through the use of ZigBee routers. In tree networks, routers move data and control messages through the network using a hierarchical routing strategy. Tree networks may employ beacon-oriented communication as described in the IEEE 802.15.4 specification. Mesh networks allow full peer-to-peer communication. ZigBee routers in mesh networks do not currently emit regular IEEE 802.15.4 beacons. Figure 3 illustrates the ZigBee Network Topology.

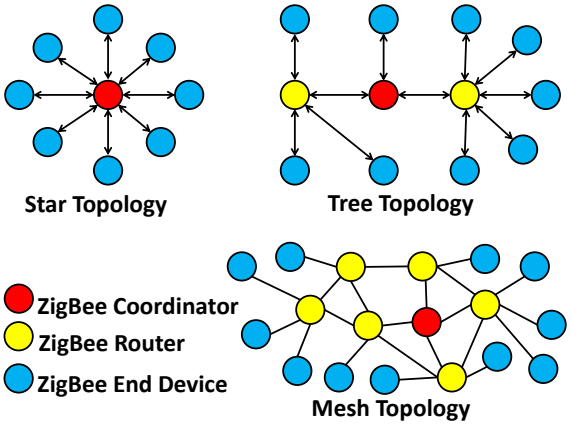

Figure 3: ZigBee Network Topology.

3.2 Attack Scenarios

As shown in Figure 1, ZigBee protocol stack consists of two main layers, the layer defined by IEEE 802.15.4 and the layer defined by ZigBee Alliance. This section outlines various attack scenarios against ZigBee devices, specifically on network layer defined by the ZigBee Alliance. Basically, ZigBee devices can be attacked to achieve four types of attacker's goals, which are:

- **Reconnaissance.** An attacker aims at gaining device information to perform further malicious activities, ranging from revealing sensitive information related to user's privacy, to gathering important information to perform follow-up attacks.
- **Denial-of-Service (DoS).** An attacker aims at performing DoS to disable the service or operations performed by the ZigBee devices.
- **Malicious Control.** An attacker aims at gaining unauthorized control to the ZigBee devices to misuse them.
- **Device Hijacking.** An attacker aims at hijacking the device connectivity to take over full control of the legitimate device. In this case, the legitimate user will lose control to their devices.

This section discusses possible known attack scenarios that could happen on ZigBee devices. The discussion mainly focuses on two attack scenarios; ZigBee network layer attacks and the possible attacks that misuses the ZigBee feature on Inter-PAN command. Figure 4 illustrates the possible attack scenarios on ZigBee devices.

3.2.1 ZigBee Network Layer Attacks

An attacker can exploit the features in ZigBee NWK layer to perform various attacks, ranging from recon-

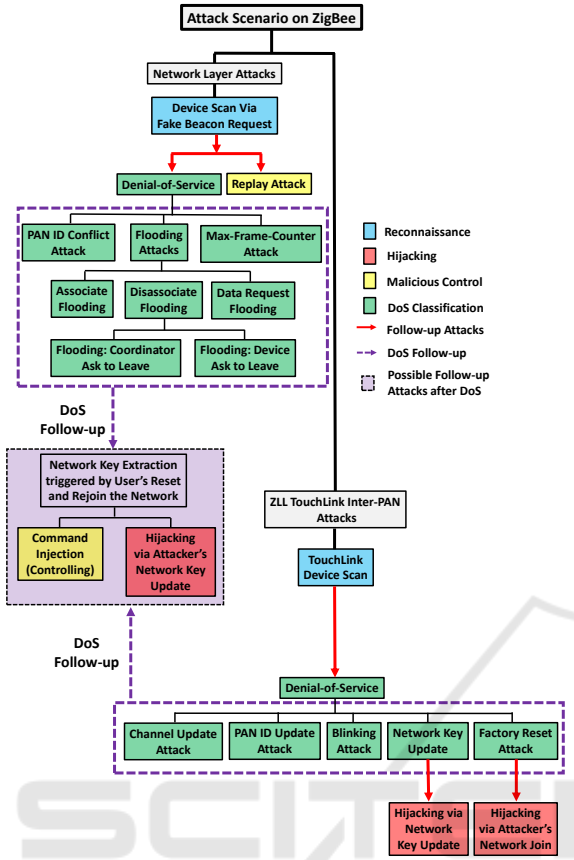

Figure 4: Attack Scenarios on ZigBee.

naissance, DoS type of attacks, malicious control, to hijacking to take control over an user's access to the target device.

(i) Reconnaissance

An attacker can scan all active ZigBee devices in the range by crafting a Zigbee standardized *Beacon_Request* packet, that we denote further as the *Fake_Beacon_Request* and broadcast it to all ZigBee channels consecutively. In this way, all coordinator and router devices in the range will answer the beacon request that could reveal sensitive information about the devices, which could be related to the privacy of the user/owner of devices. In addition, the information could also be used to perform further attacks. Figure 5 illustrates the reconnaissance in a ZigBee network.

By collecting beacon frame from coordinator and routers in the network, an attacker can get useful information, such as:

- **Channel Information.** An attacker usually broadcasts the *Fake_Beacon_Request* iteratively to all ZigBee channels (e.g. channel 11-26). In this case, the attacker can find out the channel that is being used by the target network based on the response from all devices. The channel information is useful to find out the target and perform further attacks.

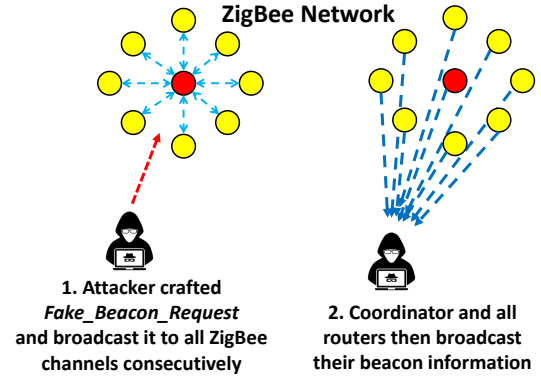

Figure 5: Reconnaissance Using Fake Beacon Request.

- **Protocol Version.** The beacon frame from ZigBee coordinator and routers contains the protocol version that is used in the network. This information is useful to initially judge what type of possible attacks could be performed on a specific version of protocol. In addition, the attacker could also infer the type of ZigBee application based on the protocol version.
- **PAN ID.** It is a ZigBee network address, which is unique for each network. PAN ID is one of important parameters that is needed to perform further attacks.
- **Extended PAN ID.** By default, Extended PAN ID (EPID) is a Coordinator MAC Address. To perform some attacks, an attacker needs to know the Coordinator's MAC Address.
- **List of All Active Devices.** To find out the target, an attacker needs information about all active devices including their short ZigBee addresses and corresponding MAC addresses.
- **Device Capability.** ZigBee beacon also contains information about the capability of each device (e.g. Coordinator, End Device, Router Capability). This information is useful to create attack scenarios on ZigBee Network.
- **Signal Strength.** This information could infer the position or distance of each ZigBee device. Thus, the attacker could get meaningful information including infer the topology of the ZigBee network.

In general, collecting ZigBee beacon of each device in the network is a critical step to performing further malicious activities.

(ii) Malicious Control

After gaining the device information, an attacker can then figure out the device functionalities in the ZigBee network, and choose a target to launch attacks such as performing malicious control. One possible way to do malicious control is by performing a Replay attack. Replay attack is a type of attack, where an attacker records or copies the previous communication packets of a legitimate device and sends the copied packets later in time. The attacker may successfully perform this attack if the legitimate device does not check the Frame Counter due to improper implementation or having flushed the last known Frame Counter from its memory. In this case, the legitimate device is made to believe that the copied packet was sent by the legitimate device in the network. For example, when a legitimate user sends a command to unlock the door, the attacker sniffs and copies the command to unlock the door and sends the copied command when the user is not at home.

(iii) Denial-of-Service

By having enough information about the target device, an attacker can perform many types of DoS attacks, ranging from PAN ID conflict attack, Max-Frame-Counter attack, to various flooding attacks.

a. PAN ID Conflict Attack

ZigBee specification defines that PAN ID shall be unique for each ZigBee network. In this way, two or more different ZigBee networks can operate in the same channel, as long as each network has a unique PAN ID. In other words, each ZigBee coordinator shall issue an unique PAN ID for its network. In addition, the ZigBee specification also defines that a node which has detected a PAN identifier conflict shall construct a *Network_Report* Command frame of type *PAN_Identifier_Conflict* and address it to the coordinator. Upon receiving the network report, the coordinator then sends a PAN ID realign to all devices in the network. The attacker can craft a ZigBee beacon frame with the same PAN ID but different Coordinator MAC address. This technique can make all ZigBee devices in the target network (e.g. the same PAN ID) to think that there is another network in the vicinity that uses the same PAN ID. All devices that receive the crafted beacon frame then construct a Network Report Command frame of type PAN Identifier Conflict. Figure 6 shows the result of our experiment, where nodes that received the specially crafted beacon frame tried to report a *PAN_Identifier_Conflict* to the coordinator.

In this test, four ZigBee nodes (i.e. *0xbcdb*, *0x95d4*, *0x59a5*, and *0xfd55*) and one coordinator

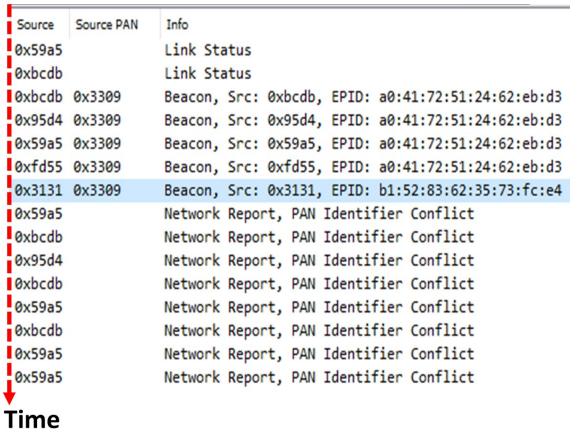

Figure 6: Network Report on PAN ID Conflict Attack.

(i.e. EPID *a0:41:72:51:24:62:eb:d3*) were set up as the legitimate network with PAN ID *0x3309*. In addition, the fake beacon frame (marked in blue) was specially crafted with node source address *0x3131*, which belongs to different coordinator (EPID *b1:52:83:62:35:73:fc:e4*), but has the same PAN ID *0x3309*. In this case, all legitimate nodes that detected the PAN ID conflict reported to their coordinator.

In PAN ID conflict attack, an attacker uses two ZigBee interfaces. The first interface is used to listen and mark PAN ID of the target network. The second interface is used to constantly send out the specially crafted fake beacon frame with the found PAN ID. These two interfaces keep listening and sending out beacons repeatedly, so that all the nodes keep asking the coordinator to change their PAN ID. Thus, the coordinator would be busy changing its PAN ID leading to a DoS of the network.

b. Max-Frame-Counter Attack

ZigBee Specification defines that a Packet Frame Counter value shall be incremented by one for each new transmission. In other words, a ZigBee device compares the frame counter value every time it receives a packet from a source address. If the received frame counter value is not higher than the previous value, the packet is then discarded by the device.

This feature introduces a type of attack called Max-Frame-Counter Attack, where the attacker crafts a spoofed packet using one of the legitimate source addresses and sets the counter to maximum value. In this case, the target device would discard any packet from the legitimate device since its frame counter is not higher than the spoofed packet that was previously sent by the attacker. An attacker may successfully perform this attack if the target device incorrectly checks the MIC (Message Integrity Code) due to improper implementation, or it could also

happen if the implementation uses a crypto-suite without integrity protection. In addition, the attackers could also successfully set the frame-counter to maximum value if he/she has access to the single network key being used.

c. Flooding Attacks
Another possible way to perform DoS attack is by continuously flooding the target device with several packets or requests. In this case, the target device would be busy responding to those requests and cannot perform its normal operation such as responding to requests from other legitimate devices. In the case of ZigBee protocol, there are several ways to perform flooding attacks, including:

- Associate Request Flooding: Associate request is a type of request where a ZigBee node initiates to join a ZigBee network. The request is sent by a ZigBee node to the network parent or coordinator. In case of flooding attack, an attacker can craft an *Associate_Request* packet and repeatedly send or flood the target network coordinator. In this case, the coordinator is flooded by many malicious requests, keeping it busy and it cannot respond to other requests from legitimate devices.
- Data Request Flooding: Data request is a type of request where a ZigBee node requests to collect data from its parent or network coordinator. An attacker can perform flooding attacks by crafting *Data_Request* and repeatedly sending the request to the target network coordinator. The attacker needs to know the network key or compromise any one node in the network to perform this attack.
- Disassociate Request Flooding: There are two types of disassociate request. The first type of request is the coordinator asking the end-device to leave the network, and the second type of request is the end-device initiating request to leave the network. An attacker could impersonate as a network coordinator and flood the target node with leave requests to make it leave the network. In similar scenario, an attacker could also impersonate as legitimate node and flood the target coordinator with leave requests. The attacker needs to know the network key to perform this attack.

3.2.2 ZigBee TouchLink Inter-PAN

ZigBee specification also defines Inter-PAN communication, which allows a node to send messages to a node in a different network (e.g. different PAN ID). In other words, this feature allows communication between networks. It is to be noted that Inter-PAN communication is not protected using NWK-layer security.

The ZigBee application profile, ZigBee Light Link (ZLL) is designed to meet the specific requirements of a connected lighting system. ZLL defines *TouchLink Commissioning* via Inter-PAN message to enable lighting use cases where commissioning of lighting devices can be done using a device with limited functionality such as handheld remote control. In addition, the *Touchlink Commissioning* also enables to manage network configurations.

Similar to Network layer attack scenarios, an attacker could exploit the features in *Inter-PAN TouchLink* commands to perform various attacks, ranging from reconnaissance to gain device information, DoS type of attacks, malicious control, to hijacking to take over user's access to their devices as shown in Figure 4. For the current work, we will focus more on the network layer attack detection.

3.2.3 Follow-up Attacks after DoS

As described before, various DoS attacks can be performed on Zigbee networks. After a successful DoS attack, where the legitimate user cannot control their devices, it is highly likely that the user would perform reset factory procedure and sequentially re-join to the legitimate network. In this regard, an attacker can exploit this user behaviour by sniffing the re-join procedure to steal and recover the key of the legitimate network. This attack can be done because the key is transported during the network join procedure and it could be encrypted using a well-known global key (such as ZigBeeAlliance09 for legacy device). Figure 7 depicts our test on stealing the key during normal network join procedure. In this test, network re-join was performed by initially sending *Associate_Request* and *Data_Request* to the coordinator/bridge (e.g. *0x01*). At the reception of those requests, the coordinator sends an *Associate_Response* to assign a ZigBee network address to the device (e.g. *0x04*) and sequentially send the network key via *Transport_Key* command. It is to be noted that for legacy devices, the *Transport_Key* is encrypted using global keys defined by the Zigbee standard. Therefore the attacker can use the recovered network key to perform further attacks including performing attacks on other legitimate device via command injection and hijacking the whole network with key updates.

(i) Command Injection
After successfully recovering the network key, an attacker can then control the target device such as impersonate as a legitimate user. In this case, the

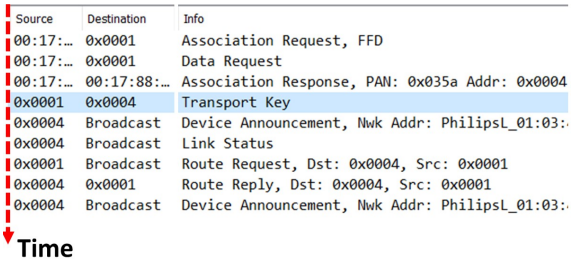

Figure 7: Sniffing the Key during Network Join Procedure.

attacker can encrypt/decrypt the communication and become a member of the legitimate network. This enables command injection sent by an attacker to be accepted by any device in the network that uses only the network key for security.

(ii) Hijacking via Network Key Update
By knowing the network key, an attacker can act as legitimate device and perform any activity that seems legitimate to other devices in the network. In this attack scenario, the attacker can impersonate as ZigBee coordinator or Trust Center and force the target node to update its network key to attacker's network key. It is done by sending the *Network_Key_Update* command to the target device. At the reception of the network key update, the target device changes its network key to the attacker's network key. It means that the device now belongs to the attacker's network and the legitimate user loses control of the target device.

4 IDS BASED ZigBee SECURITY

Intrusion Detection System (IDS) is one approach to provide reliable attack detection both for known and unknown attacks. This section outlines our proposed solution using IDS, which is specifically tailored to a large-scale ZigBee based IoT system. Initially, we describe our testbed setup used for data collection both for normal behaviour, as well as anomaly datasets created based on attack scenarios we performed on the testbed. In addition, various attack detection techniques both using rule-based and machine learning-based solution are outlined in this section.

4.1 Proof of Concept

A ZigBee IoT lighting system is used as an experimental testbed to create a realistic representative version of an IoT system. This testbed includes Zigbee based lights and a bridge, where the legitimate user can control all lights in the network using a legitimate App, such as turning on/off the lights, changing the light colour, dimming, or adding a new light bulb to join the network. It is to be noted that the testbed consists of one gateway (0x01), and three Light Bulbs (*0x02*, *0x03*, and *0x04*).

To collect realistic datasets, all ZigBee traffic is sniffed and stored to the rule engine for further analysis. To collect the normal behaviour of legitimate user, all legitimate commands sent from the App are collected via the sniffer. In this case, we use the app to repeatedly control the lights and behave like a normal user. In addition, we also performed attack scenarios on the testbed by launching various attacks described in Figure 3. By doing so, we can collect realistic malicious datasets for accurate attack detection and classification.

After collecting enough datasets at the rule engine, we then perform data analysis that is used to classify both normal and anomaly behaviour. Figure 8 depicts our lighting testbed and ZigBee IoT IDS prototype. In the prototype, we use rule engine to implement both rule-based and machine learning based detection methods.

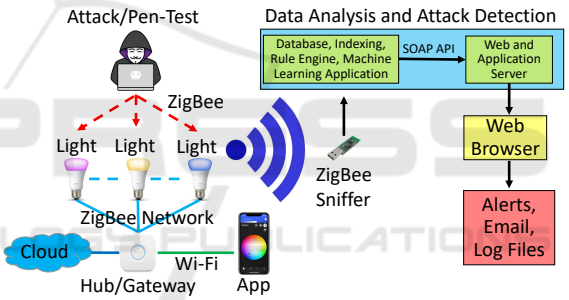

Figure 8: ZigBee IoT Lighting System and IDS Prototype.

After successfully implementing the detection methods, we tested the IDS by reproducing the attack scenarios. It is done by controlling the lighting system via the app, as well as re-launching all attacks on the testbed. In this case, the IDS prototype raises an alarm in case it detects attacks or anomalous behaviour.

4.2 Rule-based IDS

Rule-based IDS is a type of detection method using parameters based on human-crafted rules, which is created by manually analysing the dataset or log files. The rule-based IDS can be used to recognize known attacks, as well as for anomaly detection to detect unknown attacks or types of attacks that have never been seen before.

4.2.1 Attack Detection

To detect the known attacks, the dataset of various attack scenarios is analysed and compared with the dataset of legitimate user behaviour, which is sent from the App. By doing so, we can classify the normal and anomalous behaviour, as well as classify various type of attacks described in Figure 4.

(i) RSSI Pattern

By analysing the dataset, there are several possible ways to detect malicious activities. For example, malicious command injection can be detected by analysing the pattern of Received Signal Strength Indication (RSSI). In common wireless networks including ZigBee, all active devices measure the RSSI of receiving (Rx) packets including Rx unicast and Rx broadcast. The Rx RSSI sent by each neighbouring device has specific pattern depending on the distance between device sending the packet and the receiver. In addition, the RSSI pattern can vary based on external factors such as people movement or the number of people present in the vicinity of the wireless networks.

To perform command injection, an attacker initially spoofs the ID of a legitimate device. By doing so, the attacker can impersonate as legitimate device and make the target device to think that the command is sent by the legitimate device. However, the attacker cannot spoof the RSSI pattern received by the target device, meaning that there will be an unusual change in pattern of the Rx RSSI. Figure 9 depicts the detection mechanism, where the malicious command injected by the attacker changes the Rx RSSI pattern of the device *(0x03)*. In this regard, there will be a spike up on the pattern if the attacker sends the command when being closer than the legitimate device, or a spike down if the attacker sends the command from a device farther away than the legitimate device than was to the receiver. The attacker may try to send commands with a different signal strength as sent by the legitimate device. However, it would still affect the changing RSSI pattern due to external factors surrounding the legitimate and different RSSI Rx pattern being detected on other devices.

To create more accurate detection, the rule should adopt the RSSI changes based on the presence of other objects in the environment. For example, the RSSI pattern in the office during working hours is different than the RSSI pattern during night or when most of the people are not in the office (e.g. weekend or holidays). The section 4.2.2. describes in more detail the rule-based RSSI pattern anomaly detection with with the additional context of time.

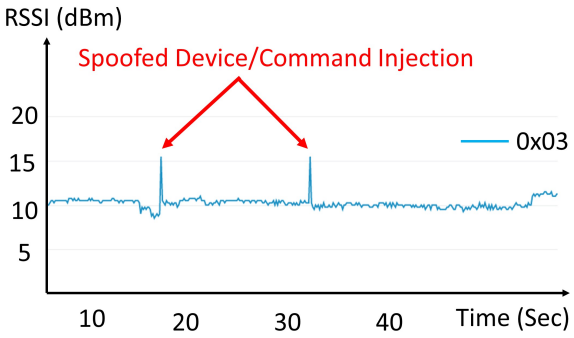

Figure 9: Command Injection and Spoofed Device Detection.

Furthermore, specific patterns of RSSI can indicate a specific attack such as various types of flooding attacks. In this case, a continuously stable RSSI value in a time range is a clear indication of flooding attacks. Figure 10 shows the attack scenario dataset for a flooding attack, where the RSSI value of a device is continuously stable in a specific period of time (e.g. 20 minutes). It is to be noted that the flooding attack was performed by initially spoofing the address of one legitimate device (i.e. *0x03*) and subsequently sending large number of packets to the coordinator.

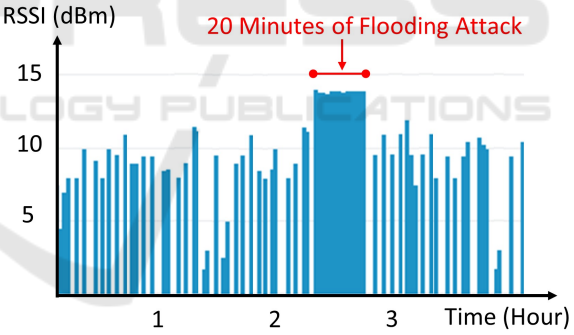

Figure 10: RSSI Pattern in Flooding Attack.

(ii) Frame Counter

The existence of a spoofed device and/or malicious command injection can be detected by analysing the frame counter of received ZigBee packets. ZigBee specification defines that the value of a packet frame counter shall be incremented by one for each new transmission by the sender. In command injection use case, an attacker attempts to impersonate as legitimate device by spoofing the ID of the device (i.e. *0x03*) and inject a message with higher frame counter to fool the target device to accept the command. However, this type of attack can be detected as the received frame counter value is significantly incremented than one. Figure 11 depicts the attack scenario dataset on com-

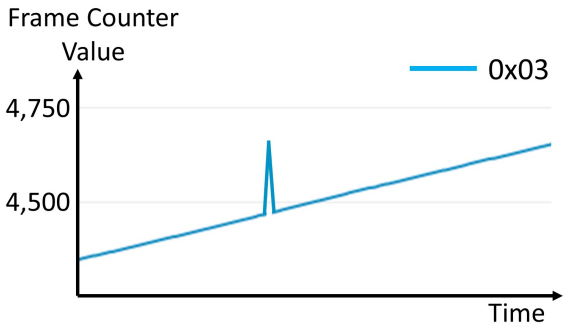

Figure 11: The Pattern of Frame-Counter Value on Command Injection Attack.

mand injection attack, where one injected packet has a significantly higher frame counter value.

Furthermore, the frame counter pattern can also indicate a specific type of command injection (e.g. various types of replay attacks). In replay attack use case, an attacker copies a previous message sent by legitimate device and replays it later to impersonate as legitimate device. However, this attack can be detected by checking the current frame counter value that is not incremented or has lower value compared to the value of previous message. Figure 12 depicts the dataset of the replay attack scenario, where the attacker repeatedly replays the pre-recorded packet sent by the legitimate device (i.e. *0x03*) which in this case is ignored by the receiving device.

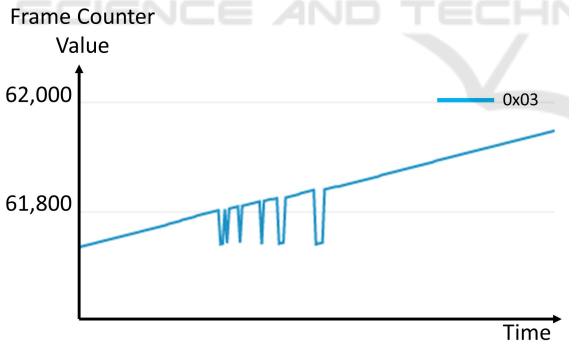

Figure 12: Frame Counter Value on Replay Attack.

(iii) Traffic Rate

Another way to detect various type of flooding attacks is by measuring the packet rate. In normal behaviour, there are certain thresholds of normal network packet exchanges among the legitimate devices. By defining the threshold, the flooding attack can easily be detected. Figure 13 shows the dataset of flooding attack with a different metric, the packet rate. It is the same flooding dataset shown in Figure 10, where the attack was performed by continuously sending large number of packets using

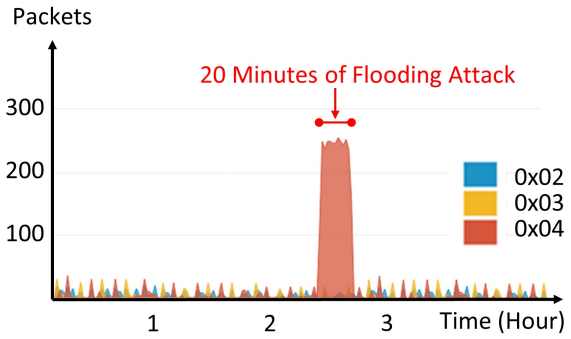

Figure 13: Dataset on Flooding Attack.

spoofed address of legitimate device (*0x04*) for 20 minutes.

(iv) Packet Frame Format

ZigBee defines various types of packet frame formats, some of them are Packet Length, Command Frames (e.g. *Route Request*, *Route Reply*, *Network Status*, *Network Report*, *Network Update*, etc.), and Frame Type. Various types of attacks in ZigBee can be detected based on specific packet frame format. For example, the existence of *Inter-PAN* command during normal operation is a clear indication of *TouchLink Inter-PAN* attacks. Therefore the IDS should raise an alarm as the *TouchLink Inter-PAN* should not exist during normal operation. Figure 14 shows the attack scenario dataset on *TouchLink Inter-PAN* attacks.

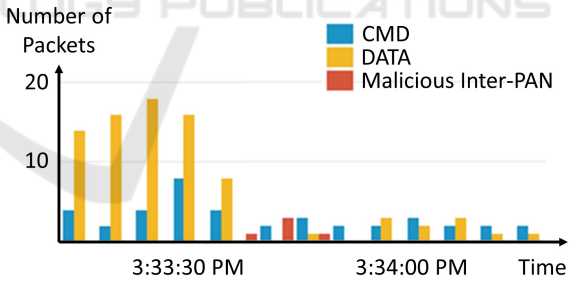

Figure 14: Dataset of Packet Frame Format.

After detecting the *TouchLink Inter-PAN* attacks, the next step is to classify the specific type of attacks, as well as approximately locate the distance of the attacker if it is applicable. It is done by checking the identifier of the TouchLink command and comparing it with the RSSI value to approximately locate the distance of the attacker. Figure 15 shows the dataset of the *TouchLink Inter-PAN Identifier* sent by the attacker. In this dataset, the attacker attempted to send several TouchLink commands, ranging from *Scan Request (0x00)* from various distances, *Reset Factory (0x07)*, *Identify Request (0x06)* for blinking attack, *Network Update (0x16)* to change the channel, to *Net-*

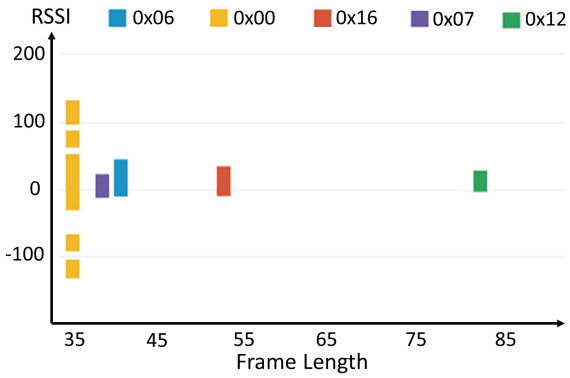

Figure 15: Distribution of TouchLink Inter-PAN Identifier.

work Join Router (0x12) to hijack the device.

By analysing the symptoms of the attacks, all known attacks basically can be detected by checking the characteristic of Packet Frame Format. For example, by checking the specific command frames that are sent during normal operation and comparing it with the specific command frames that are sent by the attacker.

4.2.2 Anomaly Detection

Rule-based method can be also used to detect unknown attacks by creating a model for anomaly detection. By using the model from the normal dataset, the IDS can detect anomaly when there is a new packet or command that does not match the rules of normal behaviours.

(i) RSSI Pattern

One method to create the rules of normal behaviour is by modelling the RSSI of normal devices. As an example of creating the models, we conducted experiments with data that was observed by a sniffer from four neighbouring nodes in a connected ZigBee lighting system in a typical office environment. The pattern of average RSSI levels of these legitimate devices are depicted in Figure 17. In this graph, the RSSI levels change based on the human presence in the office, particularly during the work hours. In this regard, the pattern of RSSI level is relatively stable on weekends as the presence of people is relatively low during those days. Figure 16 also depicts the pattern that shows RSSI is relatively stable on workdays particularly in the time range between evening hours when most people leave the office to morning hours when most people start to work on the next day.

In addition, the standard deviation of the RSSI levels over the days is also provided for a more accurate detection method. Figure 17 shows the standard deviation of RSSI levels that we observed.

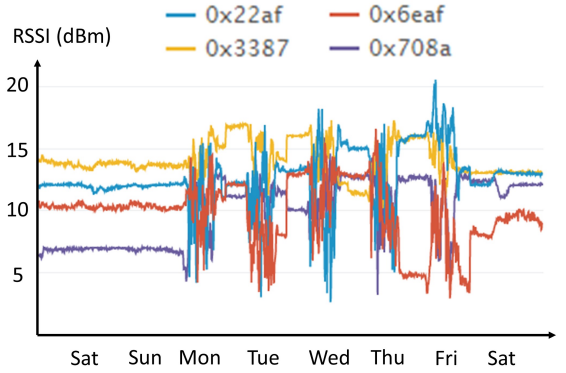

Figure 16: Average RSSI Pattern of Authorized Nodes in Every 10 Minutes.

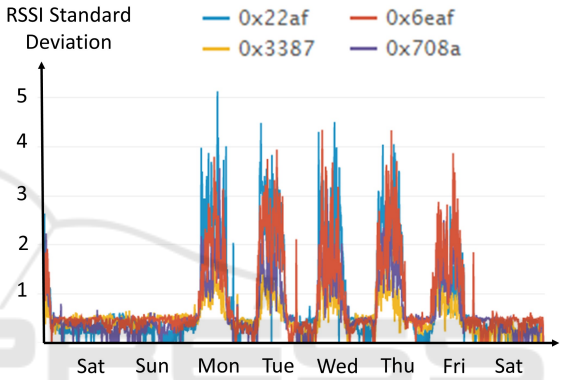

Figure 17: Standard Deviation of RSSI Pattern in Every 10 Minutes.

In this regard, the IDS will detect the existence of malicious packet or spoofed device if the current RSSI level does not match the rule model generated from the historical patterns for different time and day of the week. By doing so, all types of known attacks including spoofed device, replay attack, flooding attacks, packet injection, and in addition any types of malicious activities that have never seen before can be detected. This is because all those malicious activities (e.g. known and unknown attacks) will not match with the pattern of normal behaviour.

(ii) Packet Frame Format

A model for normal behaviour can also be created using packet frame format. There are many ways to do that, for example, by combining several features such as frame length, command frame, and the RSSI. Figure 19 depicts the model of a normal behaviour of ZigBee lighting system in an office environment. In addition, Figure 18 also identifies the anomaly command sent by the attacker, which is outside of the normal model frontier.

In this example, a model is created based on the location of the source device represented by the RSSI

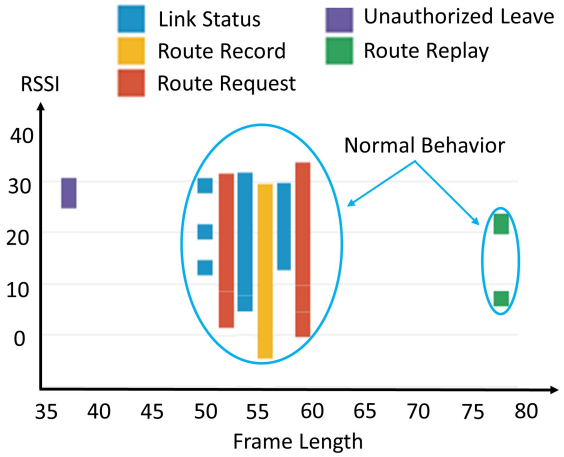

Figure 18: Anomaly Detection Using Packet Frame Format.

value, the length of the packet, and the type of command sent during the normal operation. Based on this model, an example of rules for normal behaviour can be created as follow:

- Packet length shall be either *77 bytes* or ranges from *50 to 59 bytes*.
- And the RSSI shall ranges from *-4 to 33 dBm*.
- And the legitimate command during normal operation shall be either *Link Status, Route Record, Route Request, or Route Replay*.

In general, all messages that do not match the rules above will be classified as anomaly. For example, an attacker attempting to send commands with packet length *37 bytes* and command *Leave* are classified as anomaly.

To summarize, the rule-based method is an effective method that provide detection mechanisms with high accuracy. It can be used to detect both known attacks and potential new attacks that have never been seen before. However, the human-crafted rule-based introduces complexity and is time consuming for creating the model and rules for attack detection. In addition, the rules created by humans might be error prone.

4.3 Machine Leaning Anomaly Detection

In the proposed ZigBee IoT IDS, machine learning is used to create a complex model with various features of ZigBee frame format including RSSI, frame length, time, frame counter, and packet interval. To create the model of normal behaviour, the testbed dataset of normal behaviour was used for machine learning training using One-class-SVM with non-linear kernel (RBF). Furthermore, we plotted the implementation of training/test model using scikit-learn (*sklearn.svm.OneClassSVM*). Figure 19 shows the test result of the anomaly detection model.

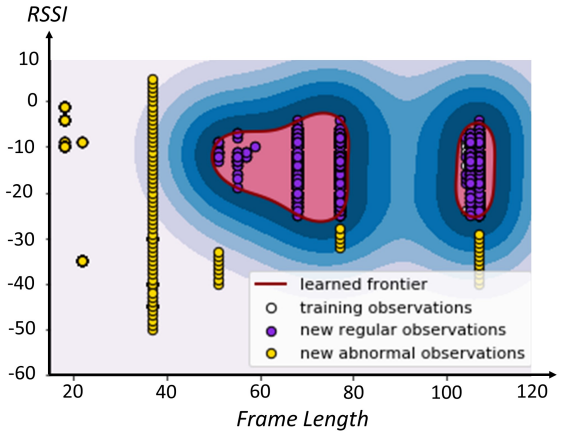

Figure 19: Training and Test Dataset Using One-Class-SVM.

To validate the accuracy of the model, we re-run the testbed to collect new normal observation (e.g. marked in purple dot) and experiment on the model by launching several attacks (marked in yellow). The attack scenarios are listed as follow:

- Flooding attack using *Leave* command: To do this experiment, we crafted the *leave* command with *37 bytes* length and flood the packet to the network coordinator from various distances. In this case, all flooded commands are spread with different RSSI ranging from *-53 to 7 dBm*. However, the *Leave* commands sent by the attacker are not inside the learning frontier, which is classified as anomaly. In addition, we crafted an *Associate_Request* command with *22 bytes* length and flood this command to the network coordinator, which is done from two different places, inside and outside the room where the testbed is installed. In this case, the plot of the commands are separated to form two different RSSI area, which are inside and outside the room. However, the command is classified as anomaly since the packet length is not in the range of packet length in normal behaviour. Furthermore, we repeated the flooding experiment using *Data_Request* command with *18 bytes* length. However, the result also shows that the command is classified as anomaly since it is outside the normal frontier.
- Finally, we attempted to perform replay attacks using previously captured packets from legitimate devices. In this case, we replayed several packets with length *51, 77, and 105 bytes*. However,

all commands are classified as anomaly (e.g. outside the normal frontier) since the attacks were sent from outside the room. It is to be noted that the attacker may attempt to perform the replay attack from inside the building, however the frame counter feature will be able to detect this attack.

In general, by combining several features to create the normal behaviour model we can detect anomalous behaviour in the ZigBee network which can be used to detect possible attempts to attack the network. In addition, using machine learning for creating the model for normal behaviour is more efficient and less-time consuming. However, there is a limitation of the non-linear kernel (RBF), where some part on the kernel (i.e. the learned frontier) touch the leaning observation (e.g. dataset of normal behaviour). This issue makes the new observation of normal behaviour to be classified as anomaly, particularly when they touch the learned frontier. Therefore, in real case IDS implementation, the One-class-SVM model will introduce false positives, where some of normal messages sent by legitimate devices will touch the frontier of normal behaviour. It means some of legitimate packets/commands will be classified as anomaly. Therefore, additional learning should be built-in to ensure the model of normal behaviour is updated correctly over time.

5 CONCLUSIONS

In this article, we have covered various methods to detect known attacks, as well as possible new types of attacks in ZigBee IoT systems. To do so, we introduce intrusion detection system with hybrid approach by combining the human-crafted rule-based and machine learning-based anomaly detection. Rule-based approach is used to provide accurate detection mechanism for known attacks, but it is also effective to be used for anomaly detection mechanism. However, the rule-based approach introduces complexity and is time-consuming to define precise rules for accurate detection. While, machine learning approach is specifically used to create a complex model of normal behaviour that is used for anomaly detection. Indeed, creating the model using machine learning-based anomaly detection is much efficient method and less-time consuming. However, it potentially introduces false alarms in real IDS deployments.

REFERENCES

A-A-Diro (2018). Distributed attack detection scheme using deep learning approach for internet of things. *Future Generation Computer Systems, vol. 82, pp. 761–768*.

Anhtuan-Le (2016). A specification-based ids for detecting attacks on rpl-based network topology. *Information, vol. 7, no. 2, p. 25*.

Chawla (2018). Security as a service: real-time intrusion detection in internet of things. *In Proceedings of the Fifth Cybersecurity Symposium, p. 12, ACM*.

Doohwan-Oh (2014). A malicious pattern detection engine for embedded security systems in the internet of things. *Sensors, vol. 14, no. 12, pp. 24188–24211*.

Granjal, J. and Pedroso, A. (2018). Intrusion detection and prevention with internet-integrated coap sensing applications. In *Proceedings of the 3rd International Conference on Internet of Things, Big Data and Security - IoTBDS,*, pages 164–172. INSTICC, SciTePress.

Maniriho, P. and Ahmad, T. (2018). Analyzing the performance of machine learning algorithms in anomaly network intrusion detection systems. In *2018 4th International Conference on Science and Technology (ICST)*, pages 1–6.

Pacheco (2016). Iot security framework for smart cyber infrastructures. *IEEE International Workshops on Foundations and Applications of Self-Systems*.

Pongle (2015). Real time intrusion and wormhole attack detection in internet of things. *International Journal of Computer Applications, vol. 121, no. 9*.

Rathore (2018). Semi-supervised learning based distributed attack detection framework for iot. *Applied Soft Computing, vol. 72, pp. 79–89*.

Saia, R., Carta, S., and Recupero, D. R. (2018). A probabilistic-driven ensemble approach to perform event classification in intrusion detection system. In *Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - Volume 1: KDIR, (IC3K 2018)*, pages 139–146. INSTICC, SciTePress.

Summerville (2015). Ultra-lightweight deep packet anomaly detection for internet of things devices. *IEEE 34th International Performance Computing and Communications Conference (IPCCC)*.

T-H-Lee (2014). A lightweight intrusion detection scheme based on energy consumption analysis in 6lowpan. *Advanced Technologies, Embedded and Multimedia for Human-centric Computing, pp. 1205–1213, Springer*.

ZigBee-Alliance (2015). Zigbee specification. *ZigBee Document 05-3474-21*.