COLIDE: A collaborative intrusion detection framework for internet of things

COLIDE: A Collaborative Intrusion Detection Framework for Internet of Things

Junaid Arshad[1], M. Ajmal Azad[2], M. Mahmoud Abdellatif[3],
M. Habib Ur Rehman[4], Khaled Salah[5]

September 25, 2018

Abstract

Internet of Things (IoT) represent a network of resource-constrained sensor devices connected through the open Internet which are susceptible to misuse by intruders. Proliferation of IoT across diverse application domains renders their security critical to ensure normal service delivery by such infrastructures. Traditional standalone intrusion detection systems are tasked with monitoring device behaviours to identify malicious activities. These systems not only require extensive network and system resources but also cause delays in detecting a malicious actor due to unavailability of a comprehensive view of the intruder's activities. Collaboration among IoT devices enables considering knowledge from a collection of host and network devices to achieve improved detection accuracy in a timely manner. However, collaboration introduces the challenge of energy efficiency and event processing which is particularly significant for resource-constrained devices. In this paper, we present an intrusion detection framework for IoT (COLIDE) that leverages collaboration among resource-constrained sensor devices and border nodes for effective and timely detection of intruders. The paper presents a detailed description of the proposed framework along with its formal description and analysis to assess its effectiveness for a typical IoT system. We implemented the COLIDE framework with Contiki OS and conducted thorough experimentation to evaluate its performance. This evaluation demonstrates efficiency of COLIDE framework with respect to energy and processing overheads achieving effectiveness within an IoT system.

1 Introduction

The use of sensor devices has increased dramatically over the last few years leading to their proliferation across diverse domains such as wearables, intelligent appliances, and vehicles. As these devices have the ability to be connected to the Internet, it introduces exciting possibilities such as the Internet of Things (IoT). IoT has received significant attention as a disruptive technology and is considered fundamental to the networks of the future. A recent study by the Gartner has predicted the number of sensor devices to increase to more than 20 billion devices by the year 2020 [1]. This has a direct impact on industrial applications such as automotive industry, commercial security cameras, as well as consumer applications such as wearables, smart TVs, and smart meters.

A typical IoT network consists of devices with resource constraints such as limited processing power, energy resources, and communication range etc. These constraints mandate an IoT network to have an efficient communication protocol that requires limited energy overheads, provides efficient performance under diverse conditions, and supports larger address space. To this extent, 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks) [2],[3],[4], allows resource-constrained sensor devices to send and receive communication events as IPv6 packets over IEEE 802.15.4 based networks. Additionally, 6LoWPAN facilitates communication among low power wireless personal area networks using IPv6 by performing header compression and fragmentation [5]. This enables the *things* to still use IP based Internet, leveraging standards and technologies developed over the last few decades. For a typical LoWPAN, this connectivity is achieved by using an edge router which facilitates connectivity among the devices participating within a LoWPAN as well as with the Internet.

However, the open network architecture of IoT has also attracted intruders to use the network of thousands of devices for spreading malicious content. Due to the proliferation of such devices in almost every aspect of our life, the threats posed due to their insufficient security are unique

with insecure devices exposing the end users to serious security and privacy threats. For instance, if an attacker is able to compromise an in-car WiFi, all in-car devices and data will be at risk. Once inside the network, an attacker can spoof the car, connect to outside data sources, and steal the owner's personal information including credit card data [6]. With regards to 6LoWPAN implementation, most of the IoT security threats originate from the 802.15.4, IP network and its adaptation layer. Therefore, the challenge for an effective and secure intrusion detection system for a 6LoWPAN based IoT network is two-fold: firstly, these devices are typically resource-constrained which limits their ability to host sophisticated security system that can monitor the device in real time. Secondly, the ad-hoc nature of 6LoWPAN networks allows devices to connect to other devices at runtime, typically for short time periods, thereby creating a dynamic network.

A number of efforts have been made to address security for IoT in general and with respect to intrusion detection in particular [7, 8, 9]. However, these are generally focused at standalone intrusion detection components which are integrated with the sensor device or the high-powered device such as *cluster head*. These approaches are limited in that they consider a restricted view of the events within an IoT network and therefore are limited in their ability to address complex, multi-stage, coordinated, and distributed attacks. We believe that collaboration in intrusion detection enables end devices to use the collective information from the number of devices to have more accurate and wider overview of the characteristics of IP traffic passing through them.

In this paper, we build on our existing work [10] and present a **COL**laborative **I**ntrusion **DE**tection (COLIDE) framework for IoT networks. In particular, the framework envisages collective use of information from the host and network-based detection systems. The detection system is divided into two layers: an edge router layer, and an end-host/node layer. The end-host component monitors the events at the node level and reports anomalous events to the network/edge router level system to correlate the alerts to perform aggregate detection. We believe correlating alerts from multiple devices will not only minimize false positive rate and improve the detection rate under distributed attacks, but will also reduce the workload at the end host. Therefore, the proposed framework is envisioned to address challenges such as the flexibility, resource constraints of the nodes, and collaborative nature of the IoT networks.

The overall contributions made by this paper are presented below.

- A novel intrusion detection framework for IoT networks focused on achieving efficient intrusion detection through collaboration between host and network-based intrusion detection.
- Efficient detection of complex, multi-stage attacks achieved via collaboration between sensor nodes and the edge router.

The rest of the paper is structured as follows. Section 2 presents the background on intrusion detection systems and 6LowPAN along with an attack model for the IoT networks highlighting specific threats addressed by the COLIDE framework. Section 3 presents the related work regarding intrusion detection within an IoT network. Section 4 presents our collaborative approach and its formal representation and analysis using Z notation. Section 5 presents the implementation of the setup that we propose followed by the performance evaluation of the proposed framework in section 6. Section 7 provides discussion of properties of proposed system. Finally, section 8 concludes the paper.

2 Background and Threat Model

In this section, we present basic concepts used throughout the rest of this paper which are important to understand the proposed system. Furthermore, we include an attack model identifying the threats addressed by the COLIDE framework.

2.1 Ipv6 over Low Power Wireless Personal Area Networks (6LoWPANs)

A major factor in the uptake of IoT is its ability to integrate sensor devices with the Internet allowing them to communicate with other devices and systems. These devices typically include automation and home appliances creating Low Power Wireless Personal Area Networks (LoWPANs).

One of the most commonly used technology for LoWPAN is IEEE802.15.4 [11]. This standard describes the PHY and MAC layer requirement for a low rate, low-power wireless embedded radio

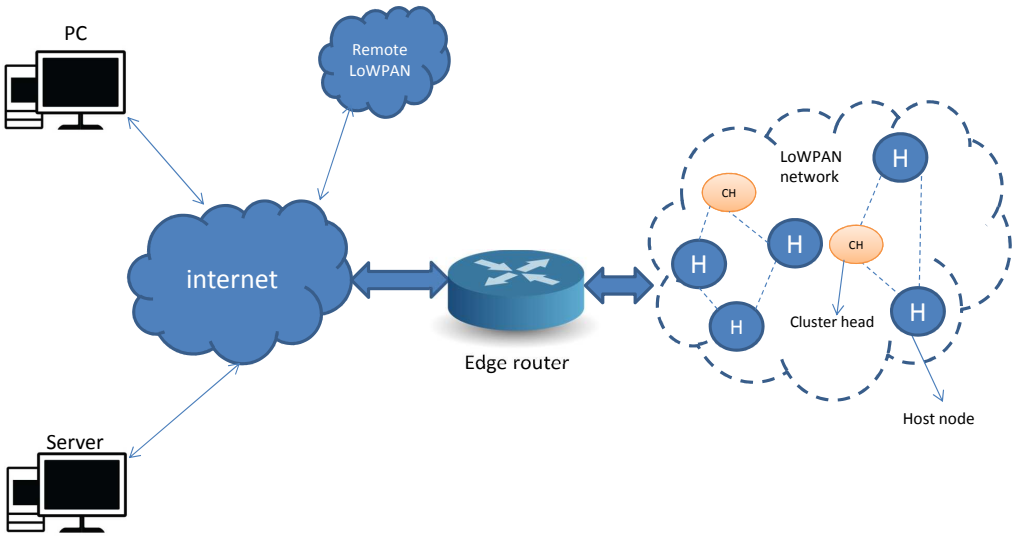

Figure 1: A typical 6LoWPAN system

communication. It is capable of operating in three different frequency band namely, *2400 MHz ISM, 915 MHz ISM and 868 MHz European band.* The MAC layer protocol is responsible for achieving efficient sharing of channel bandwidth and the quantity of energy required for efficient communication. MAC layer module controls the way packets are transmitted and received. Generally, two approaches are used in the literature to classify different types of protocol for transmission and reception of packets in the channel i.e. the reservation and the contention based protocols. The reservation-based protocols attempt to optimize energy and throughput by dividing the network into clusters referred to as Personal Area Networks (PANs). Each PAN will have a cluster head that coordinates the transmissions among the nodes within the PAN. Whereas the contention based approach uses Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) and focuses on detecting medium activity in the channel. When using CSMA/CA mechanism, a node tries to sense the medium before transmitting the packet. If another node is already sending through the medium it withdraws itself to avoid a collision when there is high traffic. In order for these devices to connect to the Internet, they each need to have a unique IP address. IPv4 has many limitations such as the address size which limits the number of connected devices. This is easily solved by using IPv6. IPv6 nodes are assigned 128 bit IP addresses in a hierarchical manner, through a network prefix of arbitrary length. IEEE 802.15.4 devices may use either of IEEE 64-bit extended addresses or, after an association event, 16-bit addresses that are unique within a PAN.

A significant problem within this context is that IPv6 requires the maximum transmission unit (MTU) to be at least 1280 octets. In contrast, IEEE 802.15.4's standard packet size is 127 octets. A maximum frame overhead of 25 octets spares 102 octets at the media access control layer. An optional but highly recommended security feature at the link layer poses an additional overhead. For example, 21 octets are consumed for AES-CCM-128 leaving only 81 octets for upper layers. In order to solve this, 6LowPAN adaptation layer was introduced.

6LoWPAN [12] is a technology standard defined by IETF to enable IPv6 stack to smoothly operate over IEEE 802.15.4 MAC link layer. As an adaptation layer, it compresses all headers including, 40 Bytes of IPv6 header from the network and 8 Bytes of UDP header from the transport into a few bytes. For IPv6 frames to be transmitted over IEEE 802.15.4 radio links, the IPv6 frames have to be divided into partitions and more data generated to resemble the original format. During packet retrieval, additional data is removed to maintain the original format. 6LoWPAN supports routing in the network and link layer.The link layer uses mesh-under while the network layer uses route-over. In mesh-under routing, the adaptation layer sends packets through multiple radio hops, while the route over scheme performs routing at the network layer with the nodes acting as a router. Thus, every hop in the link represents an IP hop to send packets across the links. Figure

1 shows the architecture of 6LoWPAN network, host devices can be either fixed (static) or mobile, depending on the application design. The edge router handles communication between 6LoWPAN devices, Internet, and other IP networks. It manages maintenance and generation of 6LoWPAN subnets and also handles data exchange between devices in the network.

2.2 Routing Protocol for Low Power and Lossy Network

As 6LoWPAN networks are expected to be densely populated. Packets will need to be routed throughout the network to reach their destination. Several routing protocols have been proposed by the 6LoWPAN community. However, only two routing protocols are currently legitimate for large-scale deployments: LOADng [13], and RPL [14]. This work is performed using Routing Protocol for Low Power and Lossy Network (RPL).

RPL is an IPv6-based Routing Protocol for LowPANs, designed by IETF Routing Over Low Power and Lossy network (ROLL) working group. It is a distance-vector routing protocol that operates on top of IEEE 802.15.4 Physical and Data Link layers. It organizes nodes in a Destination-Oriented-Acyclic-Graph (DODAG), where each router identifies a set of parents, each of which is a potential next hop on a path towards the root of the DODAG. The preferred parent is selected based on a metric or constraint among other candidates. RPL supports different kinds of network traffic, which includes; point to point, multi-point-to-point, and point to multi-point communication. RPL supports bidirectional links that enable uplink and downlink traffics. Each of the nodes in the network comprises of the Low Power and Lossy Network border router (LBR), the routers and the host. During network formation, RPL creates a tree-like topology with the border router serving as the root and the routers and host forming the edges to propagating information up and down the link of the network. Each node in RPL network has a rank, which states its position relative to other nodes with the LBR having a rank of minimum rank value, then the rank increases towards the leaves of the DODAG. The rank value is computed using the objective function. The objective function contains the routing metrics and objectives used in forming the network.

IPv6 over Low Power Wireless Personal Area Networks (6LoWPAN) [3],[4] has a profound role in this. 6LoWPAN is a networks technology that allows IPv6 packets to be carried efficiently within small link layer frames such as those defined by IEEE 802.15.4. This enables the much sought after integration of sensor devise within a WPAN with the Internet thereby realising the long-term objective of the Internet of "Things". A graphical representation of a typical 6LoWPAN is provided in Fig. 1. Typically, the Internet connectivity for the "things" is handled by a wireless access point whereas the 6LoWPAN network is connected to the IPv6 network using an edge router which handles: data exchange between 6LowPAN devices and the Internet, local data exchange between devices inside the 6LoWPAN, and generation and maintenance of radio subnet.

2.3 Threat Model for the Internet of Things

Although the IoT is an emerging paradigm, a significant part of the software stack used by the IoT applications is adopted from the existing computing paradigms. This is also evident from the integration of IoT specific stack (such as specific to 6LoWPAN and RPL) [15] with the existing Internet infrastructure such as IPv4 and IPv6. Consequently, an attack model for IoT infrastructures is not restricted to the threats specific to the new routing protocols such as 6LoWPAN and RPL but also includes threats to existing infrastructure such as IPv6, application specific attacks and attacks specific to the physical media such as the radio spectrum. The attack model for a typical IoT network is presented in Figure 2.

Our research is aimed at developing a collaborative intrusion detection system for IoT infrastructures. Therefore, we focus on two types of threats for this paper i.e. routing-specific and software or application specific threats. As our proposed system is a software entity, we render the threats at the physical layer out of scope of this research.

2.3.1 Routing-specific attacks

Routing attacks directly impact the low power devices and their routing tables. This can be achieved by making the flooding or denial of service attacks with respect to routing tables. Potential routing attacks for an IoT system are presented below.

Rank attack: 6LoWPAN networks use ranking to establish optimal routing path. Within this context, *Node Rank* indicates the quality of the path from a node to the sink node. Every time a

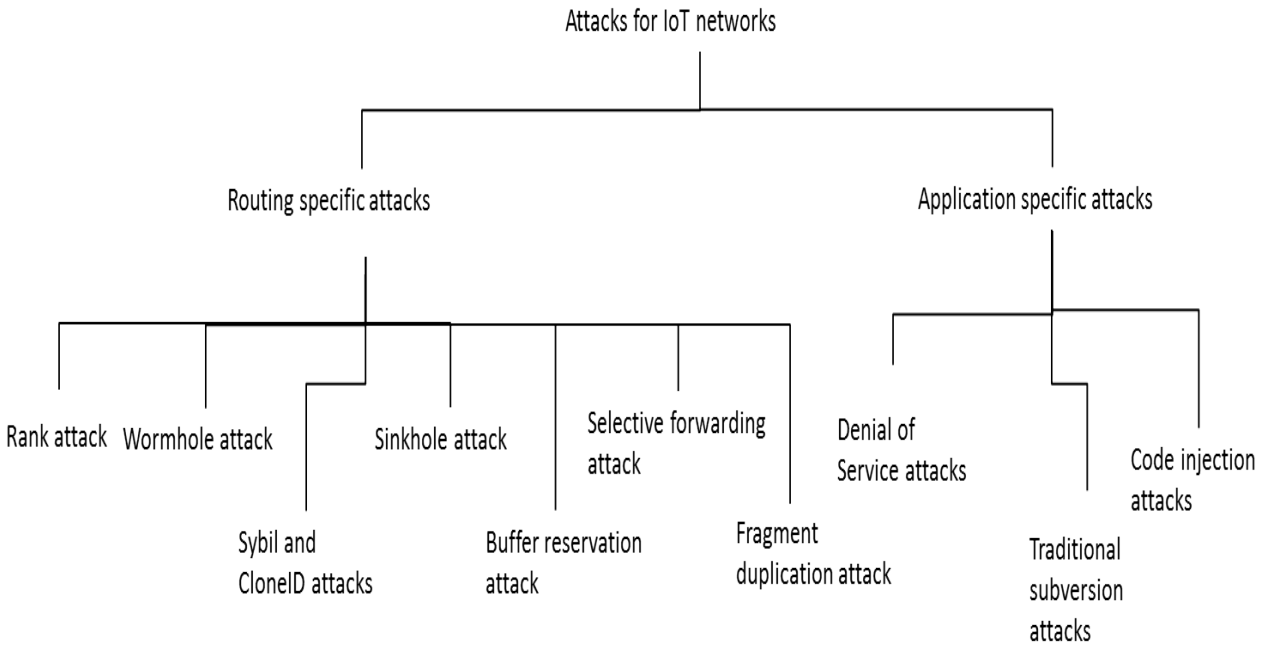

Figure 2: Attack Model for Internet of Things Network.

node updates its rank or preferred parent, it needs to inform other nodes by sending the updated information in the next cycle. RPL uses the rank rule such that a node in the parent should always have a lower rank than its children to prevent the loop creation. This enables creating an optimal topology, preventing loop creation and manage control overhead [16]. As identified by [16, 17, 18] the rank information can be maliciously tampered with by an attacker so that it chooses the node with worst rank to be its parent. This will result in disrupting the topology of the network causing delays in normal transmission.

Wormhole attack: A wormhole can be considered as a tunnel between two nodes using wired or wireless links and can be used to achieve faster transmission rates or dedicated connection between such nodes. As such, a wormhole has legitimate applications such as the connection between the local and global IDS modules within our architecture. However, a wormhole as identified by [19], can be used by an attacker to create a dedicated tunnel with a node on the Internet. Wormhole attack is not novel to the IoT networks and has been historically identified as a potential threat for wireless sensor networks[20, 21, 22].

Sinkhole attack: The objective of a sinkhole attack is to attract traffic through a designated node using illegitimate information making the node a lucrative routing sink (base station within wireless network terminology). As with wormhole attack, literature around sinkhole attack is well established with [23] being an initial effort to identify and mitigate against such attack. Creating a sinkhole does not necessarily disrupt legitimate transmission within a 6LoWPAN. However, diverting the traffic through a specific route creates opportunities to launch other attacks such as wormhole and selective forwarding attack described below.

Selective forwarding attack: With selective forwarding attack, a malicious node attempts to disrupt legitimate transmission and routing path. The malicious node, in this case, attempts to block certain packets and forward selected packets thereby affecting the routing. For instance, an attacker can forward all RPL control messages but block the rest [19]. This attack can cause more damage when used in conjunction with sinkhole attack. Such dependencies among different attack types have motivated us to explore the impact of multi-stage attacks within IoT infrastructures. To the best of our knowledge, the intrusion detection system presented in this paper is a pioneering effort to identify this issue and explore a solution to mitigate against it especially for IoT systems.

Fragment duplication attack: The fragment duplication attack leverages a weakness within the 6LoWPAN layer with respect to how fragmented packets are received and assembled by an IoT node. A consequence of the integration of 6LoWPAN with IPv6 networks is that bigger packets

supported by IPv6 have to be fragmented into smaller packets so as to be effectively processed by the resource-constrained nodes within an IoT system. However, as identified by [24], a recipient node cannot verify if two fragments of a packet were sent from the same source. Therefore, the recipient node is unable to distinguish between legitimate and spoofed fragments. A malicious node can exploit this vulnerability to block reassembly of targeted packets such as connection establishment packets. This may result in disrupting legitimate traffic as well as depleting resources available to the victim node.

Buffer reservation attack: The buffer reservation attack is closely linked to the fragment duplication attack and may be caused as a consequence of a successful fragment duplication attack. The buffer reservation attack also targets the vulnerability in the fragmentation mechanism employed by 6LoWPAN networks. As identified by [24], it leverages the fact that the recipient of a fragmented packet is unable to determine if all fragments will be received correctly. Therefore, a recipient node reserves a buffer space based on the information provided in the 6LoWPAN header with any additional fragments discarded. Taking advantage of this setting, a malicious node can send its victim single FRAG1 to reserve arbitrary buffer space thereby consuming scarce memory of the resource-constrained node.

Sybil and clone ID attack: Sybil and clone ID attacks are similar in that the objective of the attacker is to use spoofed logical identities within a network without deploying physical devices. In particular, for clone ID attack, an attacker is aiming to use a victim's logical identity within the network. Whereas, in Sybil attack, the attacker aims to assume multiple logical identities within a network without deploying physical nodes. These logical identities may not be currently present in the network. A number of existing efforts such as [19] and [23] have identified these attacks for IoT and historically for wireless sensor networks.

2.3.2 Application Specific Threats

In addition to routing specific threats mentioned above, IoT infrastructures are susceptible to other types of threats such as application specific threats. Although routing forms an essential component of the IoT system, the IoT devices are expected to run application software required by the function envisaged to be performed. We categorize these threats as application specific and present them below. *Denial of service attack:* Historically, Denial of Service (DoS) attacks are used to make the victim unavailable for legitimate service. This can be achieved via flooding the victim with the extraordinarily large volume of requests or by exhausting the resources such as memory and computational power available to the victim. Within IoT, the threat of DoS attack is two-fold: the victim can be part of the network under threat that an attacker wishes to make unavailable or the victim can be used as a zombie (stepping stone) to launch a Distributed DoS (DDoS) on a target IoT network. The significance of these threats within IoT systems have been identified by [25, 26, 27]

Malicious code injection: As identified by [26] and [28], malicious code injection is another application specific threat to IoT systems. The attacker, in this case, attempts to inject malicious code to get privileged access to the victim. Consequently, the attacker can damage the normal operation by causing a threat to the data or to the network using one of the routing specific attacks described in the previous section.

Traditional subversion attacks: In addition to the above-mentioned attacks, IoT systems are vulnerable to the existing attacks targeted at computer systems such as message interception, fabrication, modification, subversion, and phishing etc. As with the routing-specific attacks, these attacks can also form a part of a more complicated and sophisticated attack.

3 Intrusion Detection within IoT Systems

The history of intrusion detection within IoT networks has its foundations within the Wireless Sensor Networks (WSNs) where the focus has mostly been on identifying and mitigating against threats affecting routing protocols. The routing protocols within such networks were optimized to work within a resource-constrained environment and therefore prioritize performance over security [29]. With the introduction of LoWPAN and RPL networks, sensor networks are now connected to the contemporary IP network resulting in expansion of the attack surface of such networks. Therefore, such networks are not only vulnerable to malicious attempts targeting routing protocols

but also to the contemporary internet-based attacks such as code-injection, DoS, and phishing - we presented a bespoke attack model for IoT networks in the previous section. We believe, the cutting edge efforts in IDS for IoT networks should take these considerations into account to mitigate against such malicious attempts.

Within this context, we present an overview of existing efforts for intrusion detection in IoT systems. Although our research identified linkage between Wireless Sensor Networks (WSN) and IoT systems, however, we do not include literature items related to intrusion detection in WSN. This is because the type and volume of attacks faced by IoT systems are significantly different from that of WSN mainly due to IP connectivity as highlighted by the threat model presented in the previous section.

[30] presents one of the early efforts to establish an IDS for IoT where authors proposed a host-based IDS for LoWPANs using Contiki OS [14] and 6LowPAN [3],[4]. The IDS is able to perform detection based on the information at the node level and then transmit data to some centralized system for further analysis. The detection system performs detection using information collected from individual nodes and does not consider the information from other nodes in the network. The system does not show effective detection under the distributed denial of service attack that not only overwhelms the device but also congests the communication channel between nodes and the centralized system.

Kasinathan et al. [31] presented an IDS framework for 6LoWPAN which was able to detect denial of service attacks by monitoring physical parameters of the device. In [32] authors proposed a distributed system architecture for detecting the version number attacks in RPL-based networks and identify malicious nodes. Furthermore, a number of intrusion detection system architectures have been developed in [33, 34] for the resource-constrained 6LoWPAN devices based systems focusing on the sinkhole and selective-forwarding attacks (well-known attacks within 6LoWPAN networks). Our work is different from these systems, as it provides a generic framework for intrusion detection within IoT networks which is capable of working with diverse devices addressing a range of issues including different types of attacks, the inherent flexibility of the IoT networks, and the lack of trust among the participant devices.

An architecture to protect against DoS attacks within 6LoWPANs has been presented in [35] where the system uses network-level traffic and attack signature to identify malicious traffic. Moreover, Jun and Chi addressed the problem of processing a significant volume of alerts and network traffic as a part of intrusion detection in [36]. The authors identified the challenge of a significant volume of network traffic within a limited time to be processed by an intrusion detection module and proposed to use established Complex Event Processing (CEP) techniques to address this challenge. The work presented is different from our proposed system mainly due to the focus of research. For the proposed system, we focus on a generic IoT system which can include devices of any types (constrained or unconstrained) whereas the authors in [37, 31] have specifically designed the system for constrained devices. Furthermore, as a part of our proposed system, we envision to work within an untrustworthy and flexible environment where different devices can come together without previous handshakes to deliver a certain service in a coordinated manner.

A number of machine learning systems have also been proposed for detecting malicious nodes in an IoT network [38, 39, 40, 41]. However, measuring behaviour patterns of device usage and processing it via multistage neural networks could have a high-energy consumption. Additionally, the intrusion detection based on a machine learning system would require having a database of malicious and normal signatures at the IoT node, which is again not feasible due to the high memory consumption. The confidentiality and integrity of the data exchanged between IoT devices and the core system can be protected through the use of cryptography mechanism [42]. However, the energy resources required for encrypting the data makes these approaches infeasible to be deployed in a real system implementation. A number of trust and reputation based systems have also been proposed for analysing the reputation of the nodes in such network. For instance, Azad et al. [43, 44] presented a reputation aggregation system for devices in the IoT network that is then used to identify the malicious devices without affecting the privacy of the participants. Furthermore, Roux et al. [45] proposed an approach to characterize the behaviour of legitimate and non-legitimate communications based on the profiling and monitoring of the Radio Signal Strength Indication (RSSI) associated with the wireless transmissions of the connected objects. A number of trust based systems have also been proposed that compute the reputation of devices by considering the social connectivity network of devices and similarity measure between device feedback [46, 47, 48]. In [49], authors proposed a game theory based technique to perform the

anomaly detection in the IoT network by matching the attack signature. The authors further developed the reputation model based on the game theory to minimize the false positive rate incurred because of signature matching.

Our research has identified a number of existing solutions designed for resource-constrained devices i.e. RPL devices that process the packets at the centralized or local level. The solutions that are able to exploit the RPL mechanisms will be more energy efficient while providing satisfactory detection performances with respect to version number attacks. These systems normally consider the local network-agnostic view of the major events in the neighborhood devices. We believe that monitoring and intrusion detection systems should be generic and can be applied to any type of IoT devices independent of routing protocols. In view of this, the intrusion detection framework proposed in this paper does not focus on specific devices. We believe that being agnostic of the device types our proposed framework can achieve applicability in wider scenarios and application domains.

Furthermore, an IDS should take into account local and global levels of monitoring i.e. perform detection in a collaborative way. The collaborative system would allow devices to interact with each other or with a centralized system to get useful information which can be used for intrusion detection. Within this context, our proposed system implements collaboration between host and network level intrusion detection components to achieve a comprehensive view of the events. We believe this setting is paramount in enabling our proposed framework to perform efficient detection of complex multi-stage attacks.

4 The COLIDE Framework

We believe that effective intrusion detection is fundamental to achieving overall security for IoT. Within this context, we propose a collaborative intrusion detection system for IoT infrastructures (COLIDE) which takes into account unique characteristics of IoT systems and emphasizes the cooperative nature of such systems. A graphical representation of our proposed framework is presented in Fig 3.

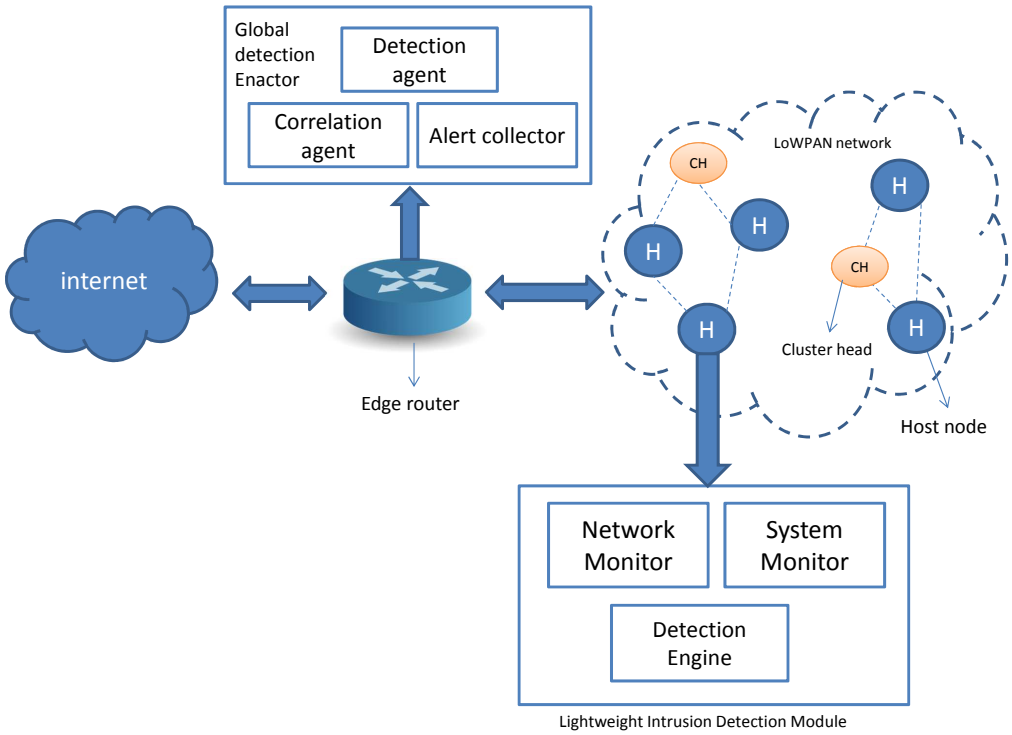

Figure 3: A collaborative intrusion detection system for IoT.

As depicted in Fig 3, the COLIDE framework proposes to conduct intrusion detection at two

levels i.e. the node, and the edge router. These are described below in more detail.

Node level detection: The node level detection module is envisioned to be a lightweight module in accordance with the relatively constrained resources available at individual sensors nodes. However, the resource-constrained nature of sensor nodes should not prohibit implementing node-level intrusion detection as the visibility offered by a node level system is unique and can improve the overall intrusion detection. In this regard, we propose using signature-based intrusion detection within nodes due to their efficiency with respect to computational resources as compared with anomaly-based detection. Furthermore, the choice of implementation for network vs host-based monitoring at node level is rendered application specific as it will influence the type of attacks that can be detected. For instance, a Denial of Service (DoS) attack targeted to flood nodes within a LoWPAN can be detected by monitoring network traffic. Whereas a backdoor channel attack targeted at gaining unauthorized access to a node can be detected by monitoring system events. Therefore, the detection engine for the node level module processes the information generated by the monitor(s) using existing signatures to detect any attack attempts.

Edge router detection: An edge router is an important component of an IoT system as it enables connectivity between the LoWPAN(s) and the Internet. With emerging infrastructures such as Fog and Edge Computing [50], the significance of edge routers has increased with multiple services such as firewalls, and traffic filters etc. being implemented at these devices. This has been possible due to increased computing capabilities of these devices. Inspired by the Fog computing paradigm, we envisage leveraging the capabilities of edge routers to achieve rigorous intrusion detection for IoT systems. To this extent, the edge router detection module is designed to monitor traffic for LoWPAN(s) attached to it, therefore, monitoring traffic for all the devices within a LoWPAN. Among others, this enables detection of attacks targeting an entire LoWPAN due to the level of visibility offered by the edge router. Within the proposed system, the edge router detection module has three components: Alert Collector, Correlation Agent, and Detection Agent.

As an edge router is expected to monitor all the sensor devices within a LoWPAN, a method is required to identify threats to individual sensors. *Alert Collector (AC)* is expected to achieve this function by communicating with individual IoT devices to gather alerts from the node level monitoring components. Due to the function of this component, an alert collector will be implemented using multi-threading as it will be communicating with multiple IoT devices simultaneously. A typical intrusion is usually not an isolated event that can be achieved within a single transaction or network event but it is usually a series of steps each of which may target a specific vulnerability with the aim to achieve the overall successful intrusion [51, 52]. *Correlation Agent (CA)* component is devised to facilitate countermeasure for such attacks by correlating malicious events at network and system levels as monitored by the node level monitors. This enables improved visibility into the events within IoT devices and facilitates the overall intrusion detection process. Historically, anomaly-based intrusion detection approaches have demonstrated better efficiency especially with respect to the detection of complex, multistage, and zero-day attacks however at the cost of increased resource consumption. Due to the increased capability of edge router devices, we, therefore, propose implementing anomaly-based intrusion detection at the edge router. The *Detection Engine (DE)* component at the edge router is expected to achieve this by making use of the alerts collected and correlated by the alert collector and alert correlation components.

In order to facilitate efficient implementation of our proposed framework and to assess its effectiveness, we have used the Z notation to represent our framework in formal representation. As described above, the COLIDE framework is composed of two major modules i.e. Node level and Edge Router module. Formal description for each of these is presented below.

4.1 Node level Module

In order to achieve a coherent representation across the different levels of security alerts, we consider the term *event* to represent an instance at the system or network level.
Let us define an event within a given host H as E_{hi}. As the proposed system is flexible in terms of implementation of the host-based component, this event can represent a system event such as a system call or a networking event such as a network packet. In both scenarios, E_{hi} will be composed of a number of parameters which will be important to decide if an event is malicious or non-malicious. For instance, for a network packet, these parameters can include protocol, inter-

arrival time, packet size etc. Therefore, E_{hi} can be represented as:

$$E_{hi} : \{p_1, p_2, p_3, \cdots p_n\}$$

where p_n is a specific parameter for an event.

Within the context of the above scenario, the Detection Engine DE is expected to categorize E_h as malicious or non-malicious. The intrusion detection policy P_h is envisioned to contribute towards this decision. Therefore, if SE_{hi} represents the state of an event E_{hi}, the following can be represented as the intrusion detection function.

$$SE_{hi} : DE(E_{hi}, P_h)$$

We define the following data models to be used throughout our formal description.
EH: a set of events for the host H
SE: state of an event; it can be malicious or non-malicious
P: Detection policy
EVENTS: a set of all the events forwarded to the CA
$\mathbf{M}_{EVENTS}$: set of malicious events forwarded to the CA
HOSTS: a set of hosts within a LoWPAN
$\mathbf{PRE}_{COND}$: a set of pre-conditions
$\mathbf{POST}_{COND}$: a set of post-conditions
DEPEND: a set of dependencies among malicious events

[Node Level Intrusion Detection]

$\Delta Host_{ID}$
$EH_{ID}? : \mathbb{N}$
$EH_i? : EH$
$PH_i? : PH$
$SEH_i? : SE$

$EH_{ID} > 0$
$EH \neq <>$
$PH \neq <>$
$SH \neq <>$
if $EH_i \in EH$ **then**
$SEH_i = DE(EH_i, PH_i)$
if $SEH_i = Malicious$ **then**
Intrusion Response(EH_i, SEH_i)

4.2 Edge Router module

The edge router module called the Global Detection Enactor (GDE) has three subcomponents i.e. Detection Agent, Correlation Agent (CA) and Alert Collector (AC). As the GDE is responsible for a LoWPAN, it is expected to monitor hosts within that LoWPAN.

Let us define $HOSTS$ as the set of hosts monitored by a GDE where H is a specific instance within this set. As described in the previous section, each host is expected to implement a node level detection engine DE_i which is tasked with categorizing events within that host as malicious or non-malicious. The AC is expected to gather this stream of events EH from the DE to be processed within the GDE. One of the significant tasks performed by the GDE is the CA i.e. to correlate multiple events to identify any multi-stage attacks. This is a complex task and is achieved by taking into account dependencies between individual events.

In order to illustrate the significance of the dependencies between different malicious events considers the following example from [53].

ME1=SadmindPing: (VictimIP, VictimPort, ExistsHost(VictimIP), VulnerableSadmind(VictimIP)),

and

ME2=SadmindBufferOverflow: (VictimIP, VictimPort, ExistsHost(VictimIP) $\wedge$ VulnerableSadmind(VictimIP), GainrootAccess(VictimIP)).

For ME_1 pre-cond can be described as:

($\{VictimIP, VictimPort\}, ExistsHost(VictimIP)$), which means that it requires a valid $VictimIP$ and $VictimPort$ to be successful in its operation i.e. ping. The $post-cond$ for the same event can be described as $\{VulnerableSadmind(VictimIP)\}$ which means that a successful ping can result in the discovery of a vulnerable service for the victim host represented by $VictimIP$. For ME_2, pre-cond can be described as $\{VictimIP, VictimPort\}, ExistsHost(VictimIP) \wedge VulnerableSadmind(VictimIP)$ which means that it requires a valid $VictimIP$, $VictimPort$ and availability of a vulnerable victim service. The $post - cond$ for this event is described as $\{GainrootAccess(VictimIP)\}$ which means that a successful buffer overflow, in this case, can result in gaining root access for the victim.

[Evaluate Dependencies]

ΔCorrelation Agent
$EH_i? : EVENTS$
$MEH_i? : M_{EVENTS}$
$H? : HOSTS$
$pre - cond? : PRE - COND$
$post - cond? : POST - COND$
$dependencies! :<>$

$MEH_i? \in M_{EVENTS}$
$M_{EVENTS} \neq <>$
$PRE - COND \neq <>$
$POST - COND \neq <>$
if $pre - cond? = post - cond$ **then**
$dependencies' = dependencies \cup MEH_i?$

The above description of the pre and post conditions for the two malicious events result in the identification of a relationship between the two seemingly distinct events. We use the above description to highlight the role of dependencies between different steps of an attack which help identifying a complex multi-stage attack. Within the context of the proposed scheme, we use the concept of dependencies among individual events within a 6LoWPAN network to detect a complex attack attempt.

Let us define EVENTS as the set of all the events monitored by CA and $M_E\,VENTS$ as the set of events identified as malicious by the node level DE. Furthermore, the pre and post conditions are represented by $[PRE - COND]$ and $[POST - COND]$, the CA populates a directed graph using the dependencies identified among the malicious events. The set of these malicious events is represented by $[DEPEND]$ in the formal system specification.

The principle followed to establish this set is borrowed from the domain of alert correlation [54] and is as follows: in order to establish these dependencies, each malicious event is considered as a tuple represented as $ME = (EH_i, pre - cond, post - cond)$. As described earlier, the parameters pre-cond and post-cond are established at the policy definition stage for intrusion detection using system call based policies. Now, for each malicious event ME_j occurring at time t_j , a malicious event ME_i occurring at time t_i is dependent on if and only if $(ME_i.post - cond = ME_j.pre - cond) \wedge (t_i < t_j)$. This establishes a directed acyclic graph where the nodes represent malicious events and the edges represent the relationship between these events.

[Detection Engine]

ΔCorrelation Agent
$MEH_i?, MEH_j? : M_{EVENTS}$
$EH_i? : EVENTS$
$dependencies - evaluation :$
$(EH_i, PRE - COND, POST - COND) \rightarrow \mathbb{P}\, DEPEND$
$d_{event} : DEPEND$

$M_{EVENTS} \neq <>$
$DEPEND \neq <>$
if $(MEH_i.post - cond = MEH_j.pre - cond) \wedge (t_i < t_j)$ **then**
Intrusion Response(MEH_i, MEH_j)

The functional description for the detection engine is explained above via the Z-notation. The dependency evaluation described earlier is the core pre-requisite for the detection engine as it identifies the linkages between different malicious events. The dataset of such dependencies is presented by the set DEPEND. Given the definitions for MEH, $post_{COND}$, pre_{COND}, t_i and t_j, the detection engine seeks to identify the correlation between the individual malicious events and executes appropriate, predefined intrusion response if the dependencies are valid.

4.3 Formal evaluation

In order to evaluate the effectiveness and feasibility of the proposed intrusion detection scheme for M2M networks, formal analysis of the scheme has been performed. The formal specification of the proposed intrusion detection scheme presented in section 4.1 and 4.2 has been used as a reference for these proofs. The formal analysis aims to assess the resilience of the proposed system to mitigate against multi-stage attacks. We prove the system resilience against multi-stage attacks.

This property represents the evaluation of the scheme with respect to mitigation against multi-stage attacks. It can be formally represented as under.

$((\exists_1 Eh_i \mid DE(Eh_i)) \wedge (\neg\Sigma DE(Eh_i))) \bullet Eh_i \in M_{EVENTS}$

In order to prove this, we can divide this into two parts. For the first part,

$\Leftrightarrow (\exists_1 Eh_i | (Eh_i)) \bullet Eh_i \in M_{EVENTS}$

From specification,

$\forall Eh_i \mid Eh_i \in M_{EVENTS} \bullet (MEH_i.post_{cond} = MEH_j.pre_{cond}) \wedge (t_i < t_j) \mid (SHE_i = Malicious \wedge SHE_i$=Malicious)

$\Leftrightarrow DE(Eh_i) \wedge SHE_i$= Malicious

$\Leftrightarrow DE(EH_i) \mid Eh_i \in M_{EVENTS} \wedge Eh_i \in DEPEND \wedge SHE_i$=Malicious

$\Leftrightarrow DE(EH_i) \mid Eh_i \in M_{EVENTS} \wedge Eh_i \in DEPEND \wedge DE(Eh_i) \mid Eh_i \in$ DEPEND

$\Leftrightarrow DE(EH_i) \mid Eh_i \in M_{EVENTS} \wedge Eh_i \in DEPEND \wedge DE(Eh_i) \in DEPEND \bullet dependencies' = dependencies \cup (Eh_i)$

$\Leftrightarrow (Eh_i.post_{cond} = Eh_j.pre_{cond}) \wedge j > i$

$\Leftrightarrow \neg(\exists_1 Eh_i \mid (Eh_i))$

$\Leftrightarrow$ false $\Leftrightarrow$ *false*

The above analysis proves that the hypothesis $\Leftrightarrow \neg(\exists_1 Eh_i \mid DE(Eh_i)) \bullet Eh_i \in M_{EVENTS}$ i.e. the intrusion detection performed at the edge router level takes into account multiple events and the dependencies among them.

Now the second part can be written as under. $(\neg\Sigma DE(Eh_i)) \bullet Eh_i \in M_{EVENTS}$

From specification,

$\forall Ehi \mid Ehi \in M_{EVENTS} \bullet (MEH_i.post_{cond} = MEH_j.pre_{cond}) \wedge (t_i < t_j) \mid (SHE_i = Malicious \wedge SHE_i = Malicious)$

$\Leftrightarrow DE(Eh_i) \bullet Eh_i \in M_{EVENTS} \Leftrightarrow DE(Eh_i) + DE(dependencies \rightarrow Indiv_{damage})$

$\Leftrightarrow DE(Eh_i) + DE(dependencies \rightarrow Indiv_{damage}) \wedge : \nu FDEPEND \bullet DEPEND \neq <>$

$\Leftrightarrow DE(Eh_i) + \Sigma MEd_i, DE(Med_i) \bullet Med_i \in dependencies \bullet DEPEND\ neq <>$

$\Leftrightarrow DE(Eh_i) \bullet Eh_i \in M_{EVENTS}$

$\Leftrightarrow$ *false*

The above analysis proves the hypothesis $(\neg\Sigma DE(Eh_i)) \bullet Eh_i \Sigma M_{EVENTS}$ to be false i.e. the intrusion detection decision evaluated by the proposed scheme does not take into account the legitimacy of each malicious event, to be false. From the above analysis, it can be concluded that the hypothesis is not valid for the proposed intrusion detection scheme. This is proved as both the conditions necessary for the hypothesis to be true have been found false in accordance with the formal specification of the intrusion detection scheme. Therefore the effectiveness of the intrusion detection scheme to correctly evaluate an intrusion whilst taking into account potential multiple stages of the intrusion is validated. However, as has been highlighted by this analysis, the correct operation of the scheme is reliant on the accuracy of the events data and the process of identifying dependencies between the individual events. Therefore, in order to ensure correct operation of the intrusion detection scheme, these aspects are required to be addressed comprehensively.

5 Experimental Setup

In this section, we describe the network setup we have used to evaluate the proposed intrusion detection system. The proposed framework was implemented with the widely used operating

system for IoT i.e. Contiki OS [55, 56]. The evaluation was conducted using Contiki v2.7 and its built-in emulator Cooja [57].

5.1 IoT Environment

The proposed IoT network is presented in Figure 4. It consists of a border router (BR) that acts as the DODAG root for the 6loWPAN network and connects it to the Internet through a SLIP interface to a computer. This computer node has higher processing power than the HOST nodes in the 6loWPAN network. The BR can also be referred to as an Edge Router.
The first tier of nodes that are a part of the 6LoWPAN are referred to in this work as routers/IDS nodes and are used to implement the host-based intrusion detection component. These nodes are responsible for forwarding messages to the root as well as periodically sending information regarding the behavior of other nodes in the network located under them in the RPL tree. In this work, the router nodes are always located one hop away from the root. The malicious node is located in the second tier and can have access to the network through one of the router nodes in the first tier. Its location can vary but in the interests of simplicity, it will always fall two hops away from the root. This is to ensure that all traffic originating from it will pass through one of the router nodes.

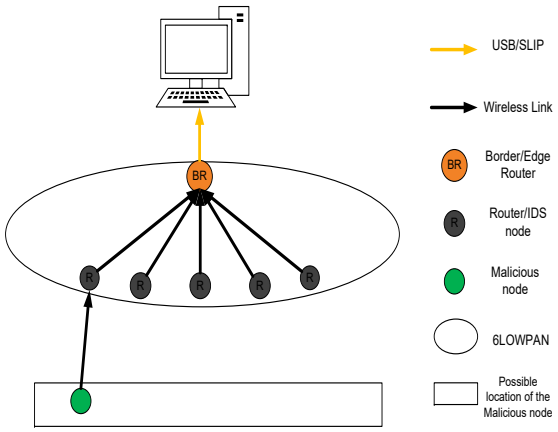

Figure 4: IDS Proposed Topology.

Simulations were performed within the Cooja Emulator with Tmote Sky Motes [58] as the chosen emulated motes. Tmote Sky node use CC2420 IEEE802.15.4 radio transceiver with 250kbps data rate operating in the 2.4GHz ISM band. Additionally, Tmote Sky has 48kb of flash and 10kb of RAM. The simulated network was created with one BR, 5 router nodes, and one malicious node. The location of the malicious node was changed between simulation runs to represent the dynamic nature of the IoT networks. However, due to the performance of RPL, the information it sends is always forwarded to its preferred parent, normally the router node closest to it. A snapshot showing the topology as created with COOJA is shown in Figure 5, where node 1 is the BR, nodes 2-6 are the IDS nodes, and node 7 is the malicious node.
We initiated the experimentation by performing a set of simulations in order to measure a baseline to be used to compare the performance of the proposed IDS system. This is achieved by simulating the network shown in the above-mentioned figure but without the presence of the malicious node. Router nodes in the baseline setup do not have any special additional code, they act as RPL routers and exchange only RPL control messages between themselves and the root. The baseline will be used to show the effect of the IDS system on the limited resources of the motes.
Another, issue that has been taken into consideration is the Radio Duty Cycle (RDC). In IoT networks, the rate of data transmission is usually low as compared to other networks. Therefore, in order to save the limited power of the nodes, it is not logical to keep the radio on whilst there is no active transmission being conducted.
This triggered many RDC protocols that are designed specifically to control the rate at which nodes can turn on or off their respective radios throughout their lifetime. In Contiki OS, the prominent RDC protocol is referred to as ContikiMAC. It takes into consideration the sleep patterns of dif-

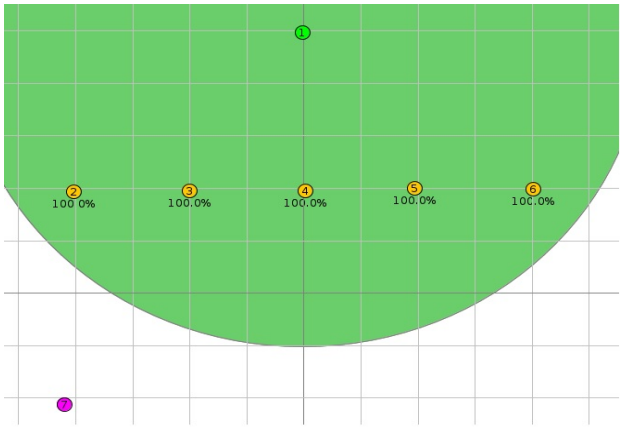

Figure 5: IDS Proposed Topology.

ferent nodes in the network when transmitting or receiving. When there is a packet to send, the node in question turns on its radio, sends the packet, waits for an acknowledgment (ACK) packet before turning off the radio and going back to sleep. If an ACK packet is not received, a retransmission cycle starts that either ends when an ACK is received or when the maximum number of retransmissions is reached. Additionally, ContikiMAC periodically turns on the radio to listen for transmissions or send updates regarding the state of the network.
Contiki OS also features an RDC protocol that keeps the radio alive regardless of active communication called NullRDC. In this paper, we have simulated the network both with duty cycling using Contikimac and without duty cycling using nullrdc. This is important as different RDC can greatly change the power consumption pattern of the node which may overshadow the effect of the IDS system.
Throughout this work, we have mainly focused on two parameters: the extra power consumption of the intrusion detection system and the extra memory footprint caused by adding the IDS features to the router nodes. These two metrics are discussed in more detail in the following subsections.

5.2 Power Measurements

As the nodes in a 6LowPAN network are usually resource constrained therefore any additional feature added to them will have to take into consideration the extra power consumption it adds to the nodes. Power measurements were made using the *powertrace* tool included in Contiki OS [59]. This tool shows the time each node spends in one of four states i.e. *transmitting* (T_x), *receiving (Rx), low power mode (LPM), and processing (CPU)*. Using these values, the energy (E) of a node that is operating with a nominal voltage (V) can be calculated by multiplying the time value for each state by the current consumption of that respective state using the following formula

$$E(mWs) = V * (T_x * 19.5 + R_x * 21.8 + LPM * 0.0545 + CPU * 1.8) \quad (1)$$

The values in the above equation are taken from the Tmote Sky data sheet and they are shown in Table 1.

Table 1: Base measurement units for Tmote-Sky nodes.

Typical Operating Conditions	**MIN**	**NOM**	**MAX**	**UNIT**
Supply voltage	2.1		3.6	V
Supply voltage during flash memory	2.7		3.6	V
Current Consumption: MCU on, Radio RX		21.8	23	mA
Current Consumption: MCU on, Radio TX		19.5	21	mA
Current Consumption: MCU on, Radio off		1800	2400	μA
Current Consumption: MCU idle, Radio off		54.5	1200	μA
Current Consumption: MCU standby		5.1	21.0	μA

The average power consumption of a single node can be calculated using the following formula.

$$Power(mW) = \frac{Energy(mWs)}{Time(s)} \quad (2)$$

Which takes into consideration the real time each node was active. Results obtained are discussed in Section 6.

5.3 RAM and ROM usage

Another scarce resource in low power networks (6LoWPAN) is the memory of the nodes. As these nodes are cheap, small, and usually expendable, they do not come with great amount of memory. As mentioned before with the tmote sky, it has only 48kb of flash and 10kb of RAM. Therefore, we measured the footprint of the code for the baseline setup and for the IDS setup to analyze the performance impact of proposed system. Results for the baseline power and energy consumptions are presented in the next section for both with and without duty cycling.

6 Evaluation

In this section, we present and evaluate the results obtained for the experimentation discussed in the previous section.

6.1 Power measurements

The experiments have been performed using COOJA emulator simulating the network presented in the previous section. For each simulation run, the malicious node sends UDP packets to the border router as its destination. Different settings of the transmission rate were tested i.e. 1, 10, 100, and 1000 packets per second. Additionally, two variations on the operation of IDS nodes were tested. The first is when the IDS node sends an update to the Border Router (BR) each time it receives 5 packets from the malicious node, referred as IDS mode5 in the figures. The second case is when the number of received packets is 10, presented as IDS mode10. This update message is a UDP message containing the source and destination IP, the source and destination port number of the malicious packets. This update message will be referred as an IDS packet. As it holds the information regarding all malicious packets sent, its size will depend on the number of malicious packets sent from the malicious node.
Figures 6 - 9 demonstrate the power consumption of the IDS node for the different scenarios simulated. The consumed power was measured after 5, 10, 15, and 20 minutes of run time with duty cycling enabled. Each plot also includes the baseline power consumptions that were measured when the nodes do not have any additional IDS component, i.e they are operating as normal RPL nodes. This was performed to provide an insight into the amount of power these nodes consume to execute basic RPL code.

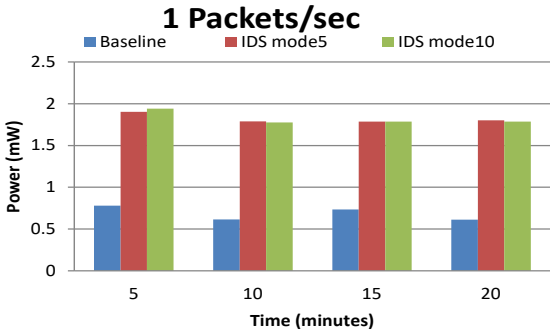

Figure 6: Power vs Time Consumption of the IDS node with duty cycling.

As can be deduced from Figures 6 - 9, the power consumption increases with the change in mode of operation for IDS node. This is because the size of an IDS packet increases when the payload increases based on number of packets from the malicious node. However, this increase is not significant to alter the overall power consumption of the system.
Another observation is that measuring the power consumption at a specific time intervals may be unfair because the number of packets sent from the malicious node is not same for each scenario. Therefore, we have also measured the power consumption at the instances after a certain number of packets from the malicious node have been received by the IDS node. This trend can be observed

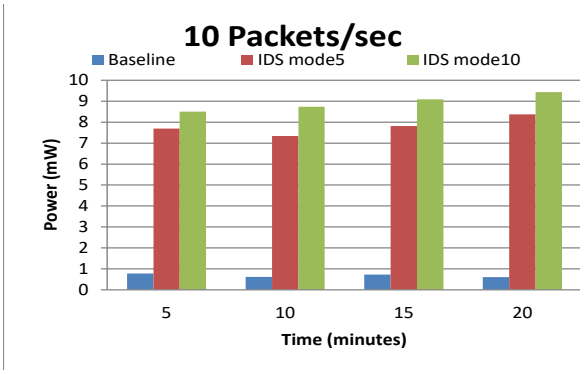

Figure 7: Power vs Time Consumption of the IDS node with duty cycling.

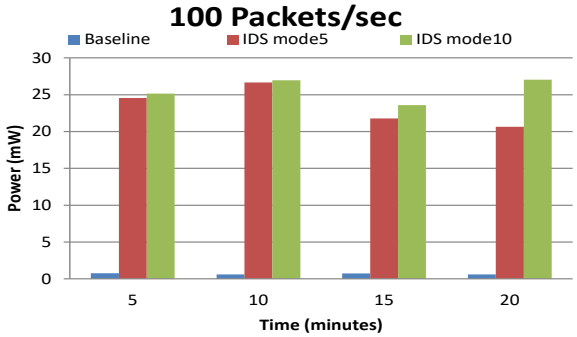

Figure 8: Power vs Time Consumption of the IDS node with duty cycling.

in Figures 10 - 13.

Additionally, we have tested the system without duty cycling, i.e. when the radio is always on and nodes never sleep. As expected, the results are almost equal for all the scenarios tested independently on the transmission rate. This is because the CPU consumes negligible power when compared with the power consumed by the radio. As the radio is always turned on, we cannot really see the difference in power consumption by the different scenarios. Nevertheless, the results where the malicious node is sending with 1 packet/sec rate are presented in the Figure 14.

6.2 Memory Overhead

As memory is an important resource on the node and therefore we have performed experimentations to assess the effect of proposed IDS component on a node. The memory footprint of the IDS module has been measured for both the baseline setup and the IDS setup. The memory overhead caused by the adding IDS functionality to the nodes is presented in Table 2.

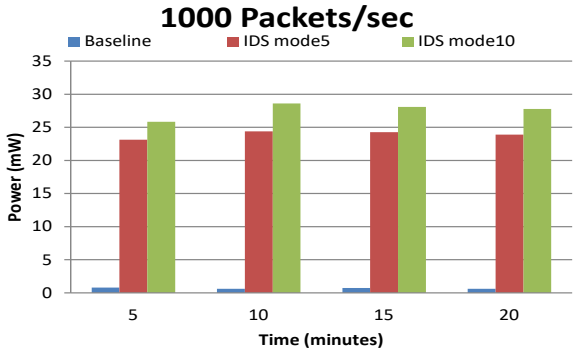

Figure 9: Power vs Time Consumption of the IDS node with duty cycling.

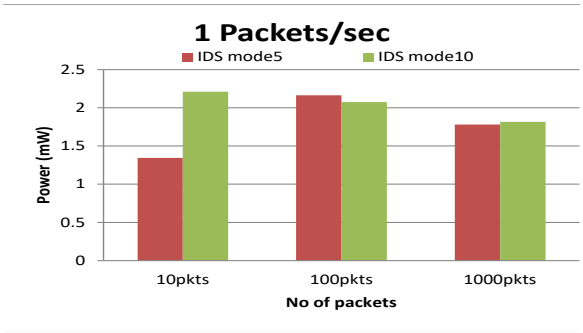

Figure 10: Power Consumption vs Number of transmitted packets of the IDS node with duty cycling.

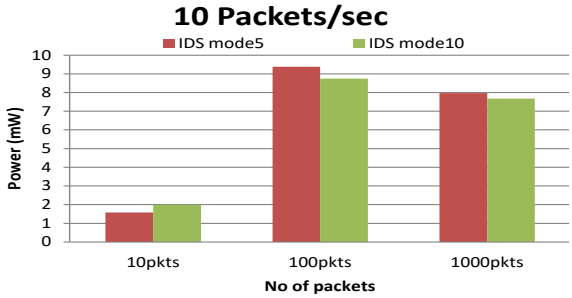

Figure 11: Power Consumption vs Number of transmitted packets of the IDS node with duty cycling.

The results of our experimentations demonstrate that packet size for the proposed IDS does not affect the ROM overhead. This is achieved as the IDS does not rely on that size. However, the RAM footprint requirements increase with the IDS packet size as more data will be saved in the memory to be sent at once. Furthermore, it is observed that the overhead is agnostic of duty cycle protocol employed as the additional application layer IDS is consistent across the underlying protocols used.

7 Discussion

In IoT networks, the IDS can be placed in two places, at the edge router or at the end host. The IDS placed at the edge router has the capability of blocking malicious traffic at the network entry point thus protect the end nodes from the malicious traffic. However, an IDS at the edge router might not consider the behaviour of the devices themselves and may lead to high communication overheads between nodes and the edge router. On the other hand, an IDS at the end nodes can monitor the performance of end nodes but it requires high processing overheads resources (processing, storage, and energy).

The existing IDSs for IoT networks are mostly isolated and monitors a single device by performing analysis for the attacks on a local device. There is no communication taking place between

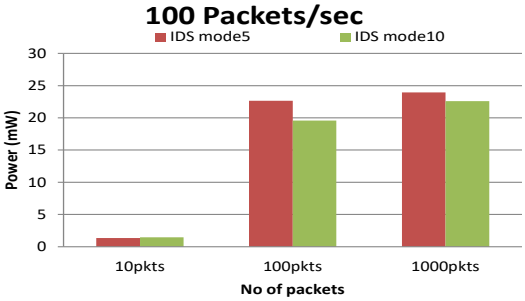

Figure 12: Power Consumption vs Number of transmitted packets of IDS node with duty cycling.

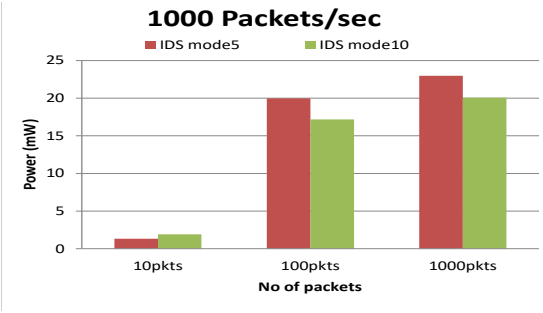

Figure 13: Power Consumption vs Number of transmitted packets of IDS node with duty cycling.

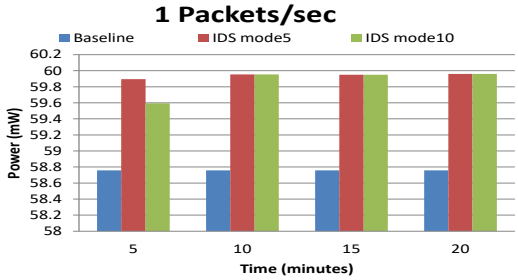

Figure 14: Power vs Time consumption for IDS node without duty cycling.

the nodes to make a collaborative detection. The standalone system will not detect most advanced and distributed attacks. Moreover, standalone systems will not be able to correlate the traffic statistics or malicious traffic passing through a number of devices at the same time. Furthermore, the standalone IDS systems always perform the detection function with respect to seen traffic on its deployed node, thus allowing the intruder to misuse the device for longer time periods. Naturally, collaboration among devices could provide the effective defense. In a collaborative system such as the one proposed in this paper, the end devices or IDS at the nodes monitor the traffic patterns and reports the events to the centralized or distributed system for event correlation and feedback aggregation. This paper represents the first attempt towards the design of collaborative intrusion detection system for the IoT network without incurring high communication or computation overheads.

Most current IDS approaches that have designed for IoT networks are mainly dependent on the underlying routing protocols (6LoWPAN and RPL). To the best of our knowledge, no approach is proposed that is generic and perform intrusion detection regardless of routing protocols. The work presented in this paper not only can be applied to a variety of routing protocols, and incorporates collaboration between nodes in the network for early and effective detection of attacks in the resource-constrained network. Our measurement results for RAM and energy consumptions reveals that the approach is lightweight and can be used in the resource-constrained set of devices. The total RAM size in the Tmote sky is 10 kb, hence COLIDE system can be deployed with the ram capacity of only 230 bytes.In terms of energy overheads, our simulation results show that the node requires only around 5mW of power for processing 1000 packets, which is negligible for the ultra-low power Tmote sky.

	Size of IDS Packets	RAM overhead (Bytes)	ROM overhead (Bytes)
With Duty Cycling	5	230	980
	10	420	980
Without Duty Cycling	5	230	980
	10	420	980

Table 2: Memory Overhead caused by the IDS functionality.

8 Conclusions

Security for IoT infrastructures is an emerging concern which requires dedicated efforts to address it effectively. This paper has focused on one specific challenge within security i.e. intrusion detection taking into account characteristics such as resource constraints of the *things* and communication among these devices. The paper has proposed a novel framework for intrusion detection which combines host and network-based detection to achieve efficient intrusion detection for IoT using 6LoWPAN.Furthermore, the paper represents pioneering effort to address the challenge of detecting multi-stage attacks for IoT infrastructures by adopting a collaborative approach. The paper has presented a detailed formal specification and analysis of the proposed system to aid its implementation in a real-life setting. The paper has also presented a detailed evaluation of the system using Contiki and Cooja simulating different scenarios to identify the overall efficiency achieved by it. Both formal and empirical evaluation of the system demonstrate its effectiveness with respect to efficient intrusion detection for IoT networks. We aim to continue this work with security evaluation of the framework such as detection accuracy and time being immediate consideration for further work.

References

[1] Gartner says 8.4 billion connected "things" will be in use in 2017, up 31 percent from 2016. [Online]. Available: http://www.gartner.com/newsroom/id/3598917

[2] H. J. W. and T. P., ",compression format for ipv6 datagrams over ieee 802.15.4-based networks," 2011.

[3] G. Mulligan, "The 6lowpan architecture," in *Proceedings of the 4th Workshop on Embedded Networked Sensors*, ser. EmNets '07, 2007, pp. 78–82.

[4] J. Olsson, "6lowpan demystified." [Online]. Available: http://www.ti.com/lit/wp/swry013/swry013.pdf

[5] G. Montenegro, N. Kushalnagar, and D. Culler, "Transmission of ipv6 packets over ieee 802.15.4 networks," *IETF RFC 4944*.

[6] D. Geer. The internet of things: Top five threats to iot devices. [Online]. Available: http://www.csoonline.com/article/2134265/network-security/the-internet-of-things–top-five-threats-to-iot-devices.html

[7] H. S. Kim, J. Ko, D. E. Culler, and J. Paek, "Challenging the ipv6 routing protocol for low-power and lossy networks (rpl): A survey," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2502–2525, 2017.

[8] G. Glissa, A. Rachedi, and A. Meddeb, "A secure routing protocol based on rpl for internet of things," in *2016 IEEE Global Communications Conference (GLOBECOM)*, 2016, pp. 1–7.

[9] A. Mayzaud, R. Badonnel, and I. Chrisment, "A Taxonomy of Attacks in RPL-based Internet of Things," *International Journal of Network Security*, vol. 18, no. 3, pp. 459 – 473,, May 2016. [Online]. Available: https://hal.inria.fr/hal-01207859

[10] J. Arshad, M. Abdellatif, M. Khan, and M. Azad, "A novel framework for collaborative intrusion detection for m2m networks," in *The 9th International Conference on Information and Communication Systems*, 2018.

[11] G. Lu, B. Krishnamachari, and C. Raghavendra, "Performance evaluation of the ieee 802.15.4 mac for low-rate low power wireless networks," in *Performance computing and communications.*, 2004, pp. 701–706.

[12] Z. Shelby and C. Bormann, "6lowpan the wireless embedded internet," *John Wiley and Sons*, 2009.

[13] T. Clausen, A. C. d. Verdiere, J. Yi, A. Niktash, Y. Igarashi, H. Satoh, U. Herberg, and C. Laven, "The lightweight on-demand ad hoc distance-vector routing protocol - next generation (loadng)," *IETF*, 2016.

[14] T. Winter, "Rpl: Ipv6 routing protocol for low-power and lossy networks," *IETF RFC 6550*, 2012.

[15] "Survey on rpl enhancements: A focus on topology, security and mobility," *Computer Communications*, vol. 120, pp. 10 – 21, 2018.

[16] A. Le, J. Loo, A. Lasebae, A. Vinel, Y. Chen, and M. Chai, "The impact of rank attack on network topology of routing protocol for low-power and lossy networks," *IEEE Sensors Journal*, vol. 13, pp. 3685–3692, 2013.

[17] W. Xie, M. Goyal, H. Hosseini, J. Martocci, Y. Bashir, E. Baccelli, and A. Durresi, "Routing loops in dag-based low power and lossy networks," in *Proceedings of 2010 24th IEEE International Conference on Advanced Information Networking and Applications*, 2010, pp. 888–895.

[18] A. Dvir, T. Holczer, and L. Butty?n, "Vera-version number and rank authentication in rpl," pp. 709–714, 2011.

[19] L. Wallgren, "Routing attacks and countermeasures in the rpl-based internet of things," *IJDSN*, vol. 9, 2013.

[20] H. Yih-Chun, P. Adrian, and B. J. David, "Wormhole attacks in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, pp. 370–380, 2006.

[21] V. Mahajan, M. Natu, and A. Sethi, "Analysis of wormhole intrusion attacks in manets," in *Proceedings of MILCOM 2008 - 2008 IEEE Military Communications Conference*, 2008, pp. 1–7.

[22] D. N. Quan and L. Louise, "A simple and efficient detection of wormhole attacks," *New Technologies, Mobility and Security*, pp. 1–5, 2008.

[23] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," *Ad Hoc Networks*, vol. 1, pp. 293–315, 2003.

[24] R. Hummen, J. Hiller, H. Wirtz, M. Henze, H. Shafagh, and K. Wehrle, "6lowpan fragmentation attacks and mitigation mechanisms," in *Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks*, ser. WiSec '13, 2013.

[25] L. O'Donnell, "Partners warn against application layer ddos attacks targeting iot devices," *Available online at: http://www.crn.com/news/internet-of-things/300084491/partners-warn-against-application-layer-ddos-attacks-targeting-iot-devices.htm?itc=refresh*, 2017.

[26] L. Chen, "Security management for the internet of things," *Electronic Theses and Dissertations*, 2017.

[27] K. Mallikarjunan, K. Muthupriya, and S. Shalinie, "A survey of distributed denial of service attack," in *Proceedings of IEEE International Conference on Intelligent System and Control*, 2016.

[28] M. Farooq, M. Waseem, A. Khairi, and S. Mazhar, "A critical analysis on the security concerns of internet of things (iot)," *International Journal of Computer Applications*, vol. 111, pp. 1–6, 2015.

[29] J. Arshad and M. A. Azad, "Performance evaluation of secure on-demand routing protocols for mobile ad-hoc networks," in *2006 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, vol. 3, Sept 2006, pp. 971–975.

[30] A. Le, J. Loo, Y. Luo, and A. Lasebae, "Specification-based ids for securing rpl from topology attacks," in *2011 IFIP Wireless Days (WD)*, Oct 2011, pp. 1–3.

[31] P. Kasinathan, G. Costamagna, H. Khaleel, C. Pastrone, and M. A. Spirito, "Demo: An ids framework for internet of things empowered by 6lowpan," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, 2013, pp. 1337–1340.

[32] A. Mayzaud, R. Badonnel, and I. Chrisment, "A distributed monitoring strategy for detecting version number attacks in rpl-based networks," *IEEE Transactions on Network and Service Management*, vol. 14, no. 2, pp. 472–486, 2017.

[33] M. Sheikhan and H. Bostani, "A hybrid intrusion detection architecture for internet of things," in *Proceedings of 8th International Symposium on Telecommunications (IST)*, Sept 2016, pp. 601–606.

[34] C. Cervantes, D. Poplade, M. Nogueira, and A. Santos, "Detection of sinkhole attacks for supporting secure routing on 6lowpan for internet of things," in *Proceedings of 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2015.

[35] P. Kasinathan, C. Pastrone, M. Spirito, and M. Vinkovits, "Denial-of-service detection in 6lowpan based internet of things," in *Proceedings of 2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct 2013, pp. 600–607.

[36] C. Jun and C. Chi, "Design of complex event-processing ids in internet of things," in *Proceedings of 2014 Sixth International Conference on Measuring Technology and Mechatronics Automation*, Jan 2014, pp. 226–229.

[37] R. Shahid, W. Linus, and V. Thiemo, "Svelte: Real-time intrusion detection in the internet of things," *Ad Hoc Networks*, vol. 11, no. 8, pp. 2661 – 2674, 2013.

[38] A. S. Obaid, S. Muhammad Shoaib, H. Choong Seon, and L. Sungwon, "Rides: Robust intrusion detection system for ip-based ubiquitous sensor networks," *MDPI Sensors*, 2009.

[39] A. Abduvaliyev, S. Lee, and Y. Lee, "Energy efficient hybrid intrusion detection system for wireless sensor networks," in *2010 International Conference on Electronics and Information Engineering*, vol. 2, Aug 2010, pp. V2–25–V2–29.

[40] L. Wenchao, Y. Ping, W. Yue, P. Li, and L. Jianhua, "A new intrusion detection system based on knn classification algorithm in wireless sensor network," *Journal of Electrical and Computer Engineering*, 2014.

[41] A. Saeed, A. Ahmadinia, A. Javed, and H. Larijani, "Intelligent intrusion detection in low-power iots," *ACM Transactions on Internet Technologies*, vol. 16, pp. 27:1–27:25, 2016.

[42] J. Granjal, E. Monteiro, and J. Silva, "Security for the internet of things: A survey of existing protocols and open research issues," *IEEE Communications Surveys Tutorials*, vol. 17, no. 3, pp. 1294–1312, 2015.

[43] M. A. Azad, S. Bag, and F. Hao, "M2m-rep: Reputation of machines in the internet of things," in *Proceedings of the 12th International Conference on Availability, Reliability and Security*, 2017, pp. 28:1–28:7.

[44] M. A. Azad, S. Bag, F. Hao, and K. Salah, "M2m-rep: Reputation system for machines in the internet of things," *Computers & Security*, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404818308666

[45] J. Roux, E. Alata, G. Auriol, V. Nicomette, and M. Kaâniche, "Toward an Intrusion Detection Approach for IoT based on Radio Communications Profiling," in *Proceedings of 13th European Dependable Computing Conference*. [Online]. Available: https://hal.laas.fr/hal-01561710

[46] Z. A. Khan and P. Herrmann, "A trust based distributed intrusion detection mechanism for internet of things," in *Proceedings of 2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*, 2017, pp. 1169–1176.

[47] Z. A. Khan, J. Ullrich, A. G. Voyiatzis, and P. Herrmann, "A trust-based resilient routing mechanism for the internet of things," in *Proceedings of the 12th International Conference on Availability, Reliability and Security*, ser. ARES '17, 2017.

[48] I. 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.

[49] H. Sedjelmaci, S. m. Senouci, and T. Taleb, "An accurate security game for low-resource iot devices," *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1–1, 2017.

[50] M. Chiang and T. Zhang, "Fog and iot: An overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, Dec 2016.

[51] F. Alserhani, M. Akhlaq, I. Awan, and A. C. MARS, "Multi-stage attack recognition system," *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications, Perth, Australia, 2010.*

[52] M. W. F. S. Cheung, U. Lindqvist, "Modelling multistep cyber attacks for scenario recognition," in *Proceedings of the 3rd DARPA Information Survivability Conference and Exposition, Washington, D.C., 2003.*, 2003.

[53] P. Ning and D. Xu, "Learning attack strategies from intrusion alerts," in *Proceedings of the 10th ACM Conference on Computer and Communications Security*, ser. CCS '03, 2003, pp. 200–209.

[54] S. O. Al-Mamory and H. L. Zhang, "A survey on ids alerts processing techniques," in *Proceedings of the 6th WSEAS International Conference on Information Security and Privacy*, ser. ISP'07. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2007, pp. 69–78.

[55] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors," in *Proceedings of 29th Annual IEEE International Conference on Local Computer Networks*, Nov 2004, pp. 455–462.

[56] Y. B. Zikria, M. K. Afzal, F. Ishmanov, S. W. Kim, and H. Yu, "A survey on routing protocols supported by the contiki internet of things operating system," *Future Generation Computer Systems*, vol. 82, pp. 200 – 219, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17324299

[57] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with cooja," in *Proceedings of 31st Local computer networks.* IEEE, 2006, pp. 641–648.

[58] "Tmote Sky," in *http://www.snm.ethz.ch/Projects/TmoteSky.*

[59] A. Dunkels, F. Osterlind, N. Tsiftes, and Z. He, "Software-based on-line energy estimation for sensor nodes," in *Proceedings of the 4th workshop on Embedded networked sensors.* ACM, 2007, pp. 28–32.