Research online Publishing
www.ronpub.com

Open Access

Open Journal of Internet of Things (OJIOT)
Volume 4, Issue 1, 2018
http://www.ronpub.com/ojiot
ISSN 2364-7108

Identifying Malicious Nodes in Multihop IoT Networks using Dual Link Technologies and Unsupervised Learning

Xin Liu, Mai Abdelhakim, Prashant Krishnamurthy, David Tipper

School of Computing and Information, University of Pittsburgh, 135 North Bellefield Avenue, Pittsburgh, USA,
{xil178, maia, prashk, dtipper}@pitt.edu

Abstract

Packet manipulation attack is one of the challenging threats in cyber-physical systems (CPSs) and Internet of Things (IoT), where information packets are corrupted during transmission by compromised devices. These attacks consume network resources, result in delays in decision making, and could potentially lead to triggering wrong actions that disrupt an overall system's operation. Such malicious attacks as well as unintentional faults are difficult to locate/identify in a large-scale mesh-like multihop network, which is the typical topology suggested by most IoT standards. In this paper, first, we propose a novel network architecture that utilizes powerful nodes that can support two distinct communication link technologies for identification of malicious networked devices (with typical single-link technology). Such powerful nodes equipped with dual-link technologies can reveal hidden information within meshed connections that is hard to otherwise detect. By applying machine intelligence at the dual-link nodes, malicious networked devices in an IoT network can be accurately identified. Second, we propose two techniques based on unsupervised machine learning, namely hard detection and soft detection, that enable dual-link nodes to identify malicious networked devices. Our techniques exploit network diversity as well as the statistical information computed by dual-link nodes to identify the trustworthiness of resource-constrained devices. Simulation results show that the detection accuracy of our algorithms is superior to the conventional watchdog scheme, where nodes passively listen to neighboring transmissions to detect corrupted packets. The results also show that as the density of the dual-link nodes increases, the detection accuracy improves and the false alarm rate decreases.

Type of Paper and Keywords

Regular research paper: *IoT, dual link technologies, malicious node identification, unsupervised learning*

1 Introduction

An IoT network typically consists of: (i) a large number of simple devices (like sensors and actuators) in the cyber-physical domain that are connected to collect and exchange information about the physical system; (ii) higher-level powerful devices in the cyber domain (like control units) that gather information and make decisions to trigger actions in the system. Distant and power-constrained IoT devices generally exchange information over multiple hops to reach to high-level controllers, constituting a multihop mesh network. The mesh topology is a flexible topology that allows any

This paper is accepted at the *International Workshop on Very Large Internet of Things (VLIoT 2018)* in conjunction with the VLDB 2018 Conference in Rio de Janeiro, Brazil. The proceedings of VLIoT@VLDB 2018 are published in the Open Journal of Internet of Things (OJIOT) as special issue.

device to communicate with any other device within its communication range, and communications to a distant receiver can be over multiple hops of transmissions. Mesh topology is adopted by many IoT protocols, such as Insteon smart home, Z-Wave, Thread and ZigBee/IEEE 802.15.4 [1][2].

However, with the increased heterogeneity and connectivity, IoT-enabled systems are vulnerable to various security threats. The risks of internal attacks launched by authenticated, yet compromised, devices increase. Devices in an IoT system could get compromised through: (i) malicious remote access over the Internet (e.g. Mirai malware [3]); (ii) malicious access to the local network (e.g. Stuxnet attack [4]); (iii) malicious physical access, especially for devices in public areas such as in smart parking infrastructures, hotels, and healthcare centers. Internal attacks launched by compromised devices could not be resolved by traditional cryptographic methods. Novel protocols and supporting architectures are needed to ensure the security of IoT systems and improve their ability to recover from attacks. Attack detection and system recovery are among the primary elements of NIST cybersecurity framework [5].

This paper focuses on packet manipulation attacks [6], which is one of the most challenging internal threats in IoT. In this attack, a compromised node along a multihop path manipulates the received information (arbitrarily or into malicious contents) before it forwards it to the destination[1]. Manipulation attacks consume network resources by having networked elements transmit/forward corrupted information; they also result in delays in decision making and could potentially lead to triggering wrong actions that disrupt the physical environment. For instance, in a healthcare application, if packets containing personal health information are manipulated by a malicious relay node, delayed, wrong or even fatal treatment decisions could be made. Similarly, manipulating information/commands sent from/to security cameras, door locks, and many other IoT elements could cause serious consequences.

Malicious packet manipulation should be detected, and nodes engaged in this activity should be identified and then removed or fixed. In a multihop network, *end-to-end packet integrity checks* with cryptographic hashes can detect packet manipulation at the destination (sink-node or gateway). Such a detection of the existence of packet manipulation would however fail *to identify which nodes are malicious* in a multi-hop transmission, since any of the relay nodes may have corrupted the packet. One approach to identifying the malicious nodes is for neighboring nodes to passively listen to transmissions and identify manipulations or packet drops. This is known by the watchdog scheme [7]. However, it implies that all nodes have to be awake and also maintain state, at least for the immediately transmitted packets. They also have to communicate their counts of potentially corrupted transmissions to the destination. The problem gets worse as the transmission paths get longer (i.e., the number of hops between the source and the destination increases). Towards solving this problem, in this paper, we propose a novel hierarchical network architecture design that utilizes two communication link technologies with distinct characteristics to facilitate inference about the trustworthiness of nodes in the network. Then, based on the dual-link enabled architecture, *we design machine learning algorithms that can effectively identify malicious nodes.* There are two broad contributions in this paper:

First, we propose a hierarchical and heterogeneous architecture that deploys trusted powerful nodes supporting dual-link (DL) technologies, referred to as DL nodes, and conventional less powerful nodes supporting a single-link (SL) technology, referred to as SL nodes. SL nodes are traditional sensors or actuators, each of which is equipped with single short-range communication interface (e.g. IEEE 802.15.4 [2]), while each of the DL nodes is equipped with long-range (LoRa) link interface [8] and short-range interface. LoRa has been recently developed targeting IoT applications. In contrast to short-range technologies, LoRa links form a star topology that can cover an entire city with a single hop, at the expense of significantly lowering the data rates. In this architecture, DL nodes use their short-range links to communicate with SL nodes, and use LoRa links to communicate with a centralized LoRa gateway (center of the star topology).

The objective of DL nodes is to evaluate the trustworthiness of SL nodes in the system. Hence, they can identify suspicious SL nodes and facilitate network recovery from attacks. To evaluate the behavior of SL nodes, DL nodes periodically exchange probe messages over their short-range links. Probe messages are propagated through multihop transmissions by SL nodes over diverse multihop network paths. By checking the integrity of packets received from each path, DL nodes compute a reputation metric of each path and a contribution metric (trustworthiness level) of each SL node along a path. The contribution metric is then used as *feature* to identify the node's behavior using K-means clustering. The distributed DL nodes make the feature calculations (trustworthiness of nodes) more accurate and the malicious node identification process

[1] Note that malfunctioning nodes could accidentally corrupt packets; hence, these nodes can also be regarded as unreliable/malicious. In the scope of this paper, any packet manipulation is regarded as malicious.

more effective. *To the best of our knowledge, we are the first to utilize dual link technologies to identify malicious nodes for network security*.

Second, we propose two methods for identifying malicious nodes, namely hard detection (HD) and soft detection (SD). Our approaches are based on *unsupervised learning and utilizing the network diversity* in different portions of the network. We proposed an earlier versions of these techniques in [9]. Here, we apply them in the hierarchical architecture supporting dual link technologies. Both techniques are based on K-means clustering. In HD, nodes are clustered into malicious or benign groups based on their contribution levels extracted at the DL nodes. In SD, nodes are clustered into three groups based on their contribution levels, then highly suspicious nodes (with very low contribution levels) are discarded and more accurate contribution feature is computed for each of the remaining nodes, provided that there is sufficient network diversity; without sufficient diversity, HD is applied instead. Unlike existing machine learning-based anomaly detection approaches that assume single-hop communication with a trusted device, such as in [10], or detect multihop attacks without identifying attackers, such as in [11], our approaches can identify malicious nodes along multihop network paths.

Our simulation results show that our approaches achieve high detection accuracy under different percentages of malicious nodes and under variable attack levels (attack probabilities) within the network. We examined the accuracy with and without channel errors. We compared our approaches with the well-known watchdog method for malicious node detection [7]. The results show that the detection accuracy of the approaches is superior to that of the existing watchdog method, and the gains increase as the percentage of malicious nodes increases. The results also show that as the density of the DL nodes increases, the detection accuracy improves and the false alarm rate decreases. The reason is that as the density of the DL nodes increases, shorter transmission paths to a DL node can be used, which enable the DL nodes to compute more accurate statistics. The long range communications here ensure that the identification results will not be manipulated in transit as it is being sent to a high level network controller (gateway) in a single hop.

2 Related Work

There are several existing techniques for detecting malicious nodes sending falsified or manipulated data in a network. In [12], an en-route filtering scheme is provided to filter false data injected by malicious nodes, where polynomials are adopted for data verification. The presented technique can detect the existence of malicious behavior in the network, but does not identify malicious nodes. A distributed detection in a centralized single hop network is considered in [13], where nodes sending falsified information to the centralized entity are detected through hard fusion rule. However, strictly centralized communication may not be available in many IoT systems with resource-constrained devices, especially as the required information rate increases. A wireless ad hoc network is considered in [14], which employs a trusted node that uses control packets, collision and channel error rates to estimate the number of packets that are maliciously dropped by its one-hop neighbors. This approach can be applied to estimate the number of maliciously manipulated packets over one hop, but could not identify malicious nodes along a multihop routing path. In [15][16], we utilized the network diversity to identify malicious relay nodes in a mesh network. The scheme requires large overhead information to be added to each packet, and can provide high accuracy under the assumption that there is at least one reliable path between the source and the destination. In this paper, we consider more general attack model and the proposed techniques reduce the amount of traffic overhead needed to identify malicious nodes.

In [7], the well-known watchdog scheme is used to identify nodes that maliciously drop packets in a multihop network. Watchdog technique relies on having every node overhear packets forwarded by neighboring nodes, and accordingly verifies whether packets were dropped. Then, nodes report to a trusted centralized entity (collector) their opinion about their neighbors' behavior. The collector uses majority voting to identify malicious nodes. The watchdog scheme can also be applied to identifying malicious nodes that manipulate packets, but it would require extra energy, computational and memory resources at each node to overhear all their neighbors' packets and evaluate their trustworthiness. In [6], a malicious node detection scheme in a tree-shaped wireless sensor network is proposed, where a sink (root of the tree) counts the percentages of manipulated packets along each path and utilizes nodes' relative position information for malicious node identification. Here, encryption is required at each node for both generating and forwarding each packet, and hence may not be supported by many of the resource-constrained IoT devices.

Machine learning methods are widely applied for detecting network attacks and malicious nodes. In [17], bandwidth and hop counts of multiple packets from a source to a destination are used as features to train a one-class support vector machine (SVM) classifier; the SVM classifier is then used to predict the existence of attacks. The scheme presented in [18] uses cross-layer features for training the SVM, which is assisted by Fisher Discriminant Analysis machine learning technique, to detect the existence of malicious behavior. Yet, malicious nodes that launched these attacks were not identified. In a multihop environmental monitoring network in [19], sensor's confidence factor is defined, which is based on the node's communication quality with its one-hop neighboring sensors; all nodes' confidence factors are collected at a control unit to train a neural network and predict data samples generated by sensor nodes, then detect anomalies. That is, if an actual data sample from certain sensor node is significantly different from the predicted data, the corresponding node is identified as faulty/malicious. In [20], Bayesian Belief Network is employed to detect outliers in a centralized network, where data is sent directly over a single hop from sensor nodes to a control unit. Both techniques in [19] and [20] rely on having each node benignly report reliable control information to a central unit, which cannot be guaranteed if some nodes are compromised. In [10], many trusted nodes are deployed in a network to have single-hop communications with ordinary nodes and send related statistical information to a control unit via secure channels, which then uses SVM to identify malicious nodes. This approach could be impractical due to the very high density of trusted nodes. In other words, it is hard to guarantee that in a mesh multihop network, each node is one-hop away from a trusted entity.

The limitation of the aforementioned techniques that are based on supervised machine learning methods is that labeling the training data can be expensive [21] or can be improperly made [22]. Unsupervised learning overcomes this problem. In [23], the unsupervised K-means machine learning method is utilized to predict anomalies. In [11], authors used a discrete time-sliding window to continuously update the feature space and an unsupervised incremental grid clustering method to identify network abnormal flows. Yet, approaches in [11][23] mainly detect abnormal flows, but do not identify compromised nodes that caused these abnormalities.

The architecture and algorithms proposed in this paper enable the identification of malicious nodes in large-scale mesh networks. With existing approaches, data collected at end-devices (servers, controllers or the cloud) carries little information about the reliability of each element along the network transmission paths. Hence, it is very hard to identify what went wrong within hidden (meshed) network connections. We develop a hierarchical architecture, where machine intelligence is applied at distributed powerful entities to accurately identify malicious nodes. The proposed architecture is described in Section 3, and the proposed malicious node identification schemes are presented in Sections 4 and 5.

3 System Model: Proposed Hierarchical Network Architecture with Dual Link Technologies

In this section, we present the architecture that utilizes dual link technologies to secure IoT networks. The main network elements are described, and then, the architecture and network operations that are followed to validate the trustworthiness of network nodes are explained. Finally, the attack model assumed in the rest of this paper is presented. Such a dual-link enabled architecture can be applied to many IoT applications to identify faults and malicious nodes within a system or between interacting systems (such as smart homes, healthcare facilities, and energy grid).

3.1 Network Elements: Dual-Link and Single-Link Nodes

We assume that the IoT network is composed of resource-constrained single-link sensors/actuators, called SL nodes, deployed in large numbers. Each SL node is equipped with a single short-range link interface (e.g., IEEE 802.15.4) supporting higher rate multi-hop transmission. More powerful nodes equipped with dual-link interfaces (DL nodes) have both long-range (e.g., LoRa) and short-range interfaces. The LoRa link at DL nodes allows the exchange of low data rate information over long distances (spanning many hops of the SL nodes). DL nodes are assumed to be trusted and communicate with a LoRa gateway in a single hop using the LoRa interface.

Both the DL and SL nodes are uniformly distributed in the network. However, the density of DL nodes is much lower than that of SL nodes. In particular, let the number of DL nodes be N_{DL} and the number of SL nodes be N_{SL}, then $N_{SL} \gg N_{DL}$. The main function of the DL nodes is to untangle hidden information in the mesh network and evaluate the trustworthiness of SL nodes. By utilizing network diversity, each DL node can compute a contribution metric (trustworthiness level) of each SL node within a predefined network region. The contribution metric is then used as input to an unsupervised machine learning algorithm to identify malicious SL nodes in the corresponding network region. The machine learning algorithms for identification of compromised SL nodes (Hard Detection and Soft Detection) are described in detail in Sections 4 and 5. The information related to identification tasks is then communicated from DL nodes to the LoRa gateway through the LoRa link. The LoRa gateway gathers information about the behavior of nodes in the entire network. Depending on the computational resources at the DL nodes, the gateway may assist in the computing

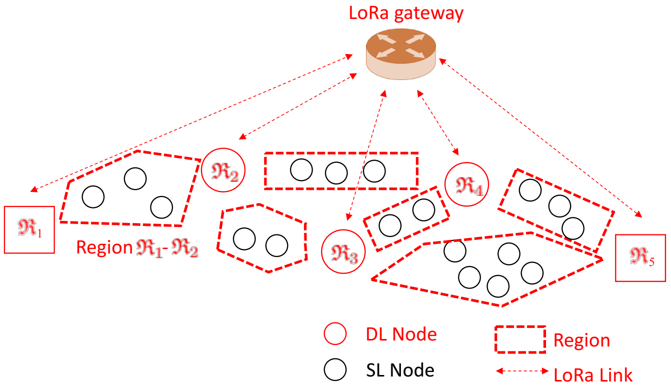


Figure 1: Large-scale network with trusted DL nodes

tasks as will be described in Section 6.

3.2 Overview of Operational Process in the Hierarchical Architecture

Consider a large-scale network shown in Fig. 1, where DL and SL nodes are uniformly distributed and DL nodes communicate with a LoRa gateway over a single hop. The j-th DL node is denoted as $\mathfrak{R}_j$. The identities of DL nodes ($\mathfrak{R}_1$, $\mathfrak{R}_2$, ..., $\mathfrak{R}_{N_{DL}}$) are revealed to each other only, and secret cryptographic keys are shared among them. The DL nodes divide the large-scale network into virtual small-scale *network portions/regions*. A region is a part of the network that includes a mesh of SL nodes and is confined by DL nodes at its farthest ends, as shown in Fig. 1. Define Region $\mathfrak{R}_j - \mathfrak{R}_k$ as a network region confined by the pair of neighboring DL nodes $\mathfrak{R}_j$ and $\mathfrak{R}_k$.

To evaluate the trustworthiness of SL nodes, DL nodes periodically exchange probe packets at random time intervals over their short-range links. Probe packets are routed through SL nodes in the corresponding region. Paths in each region are obtained using a typical route establishment phase, which allows DL nodes to gather information about the network topology. The probe packets are flooded in each region (i.e., transmitted over many transmission paths in that region) to validate the integrity of intermediate SL nodes. We note that flooding in packet routing is used in some IoT protocols for local or personal area networks, such as Z-Wave and Bluetooth mesh. In this paper, flooding is made over a network region and not over the entire network to minimize the traffic. We assume that DL nodes can get information about the route taken by each probe packet they receive.

When a DL node receives probe packets from diverse routing paths within a region, it examines these packets and evaluates a contribution metric (trustworthiness level) for each SL node that assisted in relaying the packet. The contribution metric is then used as the *feature* for identifying compromised SL nodes, as will be described in Section 4. Note that other than probe packets, transmission of a packet over the network can be over a single routing path (e.g. using ad-hoc on-demand distance vector routing). In this case, DL nodes may also act as intermediate check-points that monitor general features about the traffic to detect anomalies. However, in this paper we focus only on identifying malicious SL nodes using the probe packets.

3.3 Attack Model

We assume that (the few) DL nodes are trusted (and can be perhaps manually audited), while (the many) SL nodes are simple resource-constrained devices that could get compromised. In this paper, we will focus on manipulation attacks, where a compromised device manipulates a packet before it forwards it through multiple hops towards the destination. The attack model is assumed to be static, where a malicious node manipulates each packet it forwards by a fixed probability. A malicious node's behavior is independent of the routing path. This assumption is reasonable as the resource constrained nodes are unlikely to have the ability to respond intelligently to detection approaches even when compromised.

Let R_i be an SL node, and define α_i as a binary flag to express whether R_i is benign. If R_i is a malicious node, it manipulates each packet it forwards by a fixed probability P_i. $\bar{P}_i = 1 - P_i$ is the probability that a forwarded packet will not be manipulated. If R_i is benign, then $P_i = 0$. That is,

$$\alpha_i = \begin{cases} 1, & \text{if } R_i \text{ is benign,} \quad P_i = 0, \\ 0, & \text{if } R_i \text{ is malicious, } 0 < P_i < 1. \end{cases} \tag{1}$$

Table 1: Notations

$\mathfrak{R}_j$	j-th DL node in large-scale network
S	Source for a region
D	Destination for a region
N	Total number of SL nodes in a region
R_i	i-th SL node in a region
P_i	Attack probability of i-th SL node in a region
$\mathfrak{l}_i$	i-th path in a region
$\mathcal{T}$	Path reputation value set for a region
T_i	i-th path's reputation value in a region
$\mathcal{C}$	Node contribution value set for a region
C_i	i-th node's contribution value in a region
$\mathcal{G}_B^{(X)}$	Benign node set for a region by X algorithm (X can be HD, SD, ESD)
$\mathcal{G}_M^{(X)}$	Malicious node set for a region by X algorithm (X can be HD, SD, ESD)

Table 2: Abbreviations

DL node	A DL node has dual communication links, is more powerful and trusted
SL node	A SL node has single communication link, is less powerful and can get compromised
HD	Hard Detection algorithm
SD	Soft Detection algorithm
ESD	Enhanced Soft Detection algorithm

4 Small-Scale Network Model and Hard Detection

Now, we describe the malicious node identification methods that are carried out in each network region. In this section, first we define the contribution feature through which the nodes are clustered, then present the hard detection (HD) malicious node identification scheme (the soft detection scheme will be explained in Section 5). The algorithms are based on K-means clustering. Important notations and abbreviations are listed in Table 1 and Table 2, respectively. We focus on a single Region $\mathfrak{R}_j - \mathfrak{R}_k$ where $\mathfrak{R}_j$ acts as the source S and $\mathfrak{R}_k$ acts as the destination D in this region. Analysis can be applied to each region in a similar manner. Denote N as the total number of SL nodes assisting the multihop packet transmissions between S and D. Let $R_i, i \in [1, N]$ be the i-th SL node in the small-scale network between S and D. In general, there could be multiple available routing paths from S to D and each path may contain different number of relay nodes (SL nodes). Each node (including S and D) has a short-range RF link with communication range of radius r. S sends probe packets over multiple joint routing paths to D. That is, when S sends one probe packet, D can receive a copy of this packet from each of the available paths. In this section, we assume no channel errors. We will discuss how to involve channel errors in the next section.

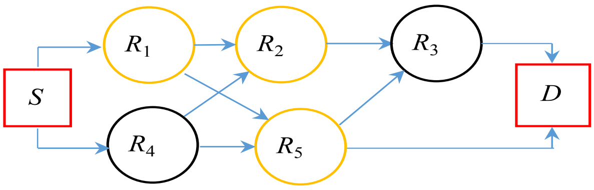


Figure 2: A mesh-like multihop network example with different path lengths

4.1 Path Reputation and Node Contribution

The goal here is to identify malicious nodes within a network region. For this purpose, we calculate path reputation metric and node contribution levels (trustworthiness). At the destination DL node D in a network region, among all the received probe packets along a certain path, there could be some packets manipulated by compromised nodes. D estimates the number and percentage of (un)manipulated packets along each path by checking the integrity[2] of each received packet using a keyed hash function. The higher the percentage of correctly received packets is, the higher the reputation the corresponding path has for delivering packets. Thus, we define a path's reputation as the number of correctly received packets going through a path divided by the number of all packets transmitted through it. Let all the L paths in a network region be in the path set $\mathcal{L} = \{\mathfrak{l}_i\}, i \in [1, L]$. Each $\mathfrak{l}_i$ represents a possible S-D path, i.e., group of relay nodes that forward packets between S and D. For example, in Fig. 2, the path "$R_1 - R_2 - R_3$" is represented as $\mathfrak{l}_i = \{1, 2, 3\}$.

The reputation value of path $\mathfrak{l}_i$ is denoted by $T_{\mathfrak{l}_i}$. High $T_{\mathfrak{l}_i}$ indicates that more packets along $\mathfrak{l}_i$ were received correctly. We can then quantify each node's contribution to a path's reputation. Let's consider the mesh-like multihop network in Fig. 2 as an example, where there are total of five relay nodes and multiple possible paths between S and D. We assume that R_1, R_2 and R_5 are malicious nodes (marked in yellow color). Each available direct communication link is denoted by an arrow. Define $\mathcal{M}_{\mathfrak{l}_i}$ as the set of manipulated packets along the path $\mathfrak{l}_i$ and $\bar{\mathcal{M}}_{\mathfrak{l}_i}$ as the set of correctly received packets. First, let's focus on path "R_1-R_2-R_3", through which S sends Q probe packets. $\mathcal{M}_{1,2,3}$ is the set of

[2] To achieve the integrity check, each packet at S contains a message concatenated with a corresponding hash value; D checks the integrity using a cryptographic keyed hash function where the key is shared by S and D.

the manipulated packets along this path ("R_1-R_2-R_3"), $E[|\mathcal{M}_{1,2,3}|]$ is the expected number of manipulated packets and $E[|\bar{\mathcal{M}}_{1,2,3}|]$ is the expected number of correctly received packets. Recall the attack model in Section 3.3 that R_i manipulates packets with probability P_i; then $E[|\bar{\mathcal{M}}_{1,2,3}|] = \bar{P}_3\bar{P}_2\bar{P}_1 Q$. Therefore, the expected reputation of path "R_1-R_2-R_3" becomes

$$\bar{T}_{1,2,3} = E[|\bar{\mathcal{M}}_{1,2,3}|]/Q = \bar{P}_3\bar{P}_2\bar{P}_1 = \prod_{i=1}^{|\{1,2,3\}|} \bar{P}_i. \quad (2)$$

$T_{\mathfrak{l}_j}$ is the estimate of $\bar{T}_{\mathfrak{l}_j}$ obtained by sufficiently large number of probe packets. We denote the i-th node's contribution to $T_{\mathfrak{l}_j}$ as $C_i^{\mathfrak{l}_j}$. It indicates how R_i contributes to the reputation of path $\mathfrak{l}_j$ by benignly forwarding packets, i.e., $C_i^{\mathfrak{l}_j}$ represents the trustworthiness of R_i along path $\mathfrak{l}_j$. For each path $\mathfrak{l}_j$, since there is no prior information about P_i, our approach is to initially assume that each node has equal contribution to the reputation of its path. From Eq. (2), the equality is expressed by $T_{\mathfrak{l}_j} = (C_i^{\mathfrak{l}_j})^{|\mathfrak{l}_j|}$. That is, $C_i^{\mathfrak{l}_j} = \sqrt[|\mathfrak{l}_j|]{T_{\mathfrak{l}_j}}$. For example, in Fig. 2, for path "R_1-R_2-R_3", the contribution value at each node is $C_i^{1,2,3} = \sqrt[3]{T_{1,2,3}}$, $i = 1,2,3$. Since each node is associated with multiple paths, node's contribution value in other paths should also be taken into account. Let the node R_i be associated with a total of k_i paths, and $k_{i,j}, j \in \{1,...,k_i\}$ is the j-th path among these k_i paths. Path $k_{i,j}$ contains $|k_{i,j}|$ relay nodes. The reputation of path $k_{i,j}$ is denoted as $T_{i,j}$. Then R_i's overall contribution value is calculated as

$$C_i = \frac{1}{k_i}\sum_{j=1}^{k_i} \sqrt[|k_{i,j}|]{T_{i,j}}. \quad (3)$$

For example, in Fig. 2, R_1 is associated with a total of k_1 = 3 paths. $k_{1,1} = \{1,2,3\}$ ($|k_{1,1}| = 3$) denotes the path "R_1-R_2-R_3", $k_{1,2} = \{1,5\}$ ($|k_{1,2}| = 2$) denotes the path "R_1-R_5" and $k_{1,3} = \{1,5,3\}$ ($|k_{1,3}| = 3$) denotes the path "R_1-R_5-R_3". R_1's contribution values in $k_{1,2}$ and $k_{1,3}$ are $C_1^{1,5} = \sqrt[2]{T_{1,5}}$, $C_1^{1,5,3} = \sqrt[3]{T_{1,5,3}}$, respectively. R_1's overall contribution value is the average of R_1's contribution values in its three associated paths, which is expressed as $C_1 = (\sqrt[3]{T_{1,2,3}} + \sqrt[2]{T_{1,5}} + \sqrt[3]{T_{1,5,3}})/3$.

4.2 Hard Detection

Generally, if the attack probability of R_i (P_i) is relatively high, i.e., R_i manipulates packets going through it with high probability, then R_i's associated path reputations are relatively low and its contribution value C_i will also be low. On the contrary, benign nodes (with $P_i \simeq$ 0) will have relatively high contribution values. The contribution value/metric is used as feature to classify nodes' behavior. We utilize the K-means clustering algorithm, which can cluster nodes into multiple groups according to their similarity [24]. Denote the contribution dataset as $\mathcal{C} = \{C_i\}, i \in \{1,...,N\}$. K-means method clusters the nodes in two groups; nodes with *similar contribution values* are clustered in the same group. The group of nodes with higher contribution values is identified as the benign node set $\mathcal{G}_B^{(HD)}$ and the other group becomes the malicious node set $\mathcal{G}_M^{(HD)}$. We call this algorithm Hard Detection (HD) as nodes are classified into two groups. The steps of this algorithm are described in Algorithm 1[3].

Algorithm 1 Hard Detection: HD($\mathcal{C}, \varepsilon, \mathcal{L}$)

```
Input dataset $\mathcal{C}$, threshold $\varepsilon$, path set $\mathcal{L}$;
if $\mathcal{C} = \bar{1}$ then
    $\mathcal{G}_B^{(HD)} = \{1,...,N\}, \mathcal{G}_M^{(HD)} = \varnothing$;
else if $\tilde{T} < \varepsilon$ then
    $\mathcal{G}_M^{(HD)} = \{1,...,N\}, \mathcal{G}_B^{(HD)} = \varnothing$;
else
    Use K-means($\mathcal{C}$,2) to cluster $\mathcal{C}$ into 2 groups. The group with higher data values is $\mathcal{G}_B^{(HD)}$; the other group is $\mathcal{G}_M^{(HD)}$;
end if
Output benign node set:$\mathcal{G}_B^{(HD)}$; malicious node set:$\mathcal{G}_M^{(HD)}$.
```

Note that there are two special cases before applying the K-means method: First, if all the nodes are benign, then all the path reputations are high (equal to 1) and each node's overall contribution value is 1; in this case, we identify all nodes as benign. Second, if all nodes are malicious, most paths' reputation values will be very low (close to 0). In our approach, if the average of all paths' reputation values is less than a threshold ε (a small value), then all nodes are identified as malicious. The average path reputation value is calculated as $\tilde{T} = \frac{1}{L}\sum_{i=1}^{L} T_{\mathfrak{l}_i}$.

5 Soft Detection and Channel Errors

5.1 Soft Detection

As explained in the previous section, nodes with relatively high P_i are expected to have relatively low contribution values, and nodes with P_i = 0 (benign nodes) are expected to have high contribution values. However, benign nodes with low P_i may have their

[3] Note that K-means($\mathcal{A}, n$) refers to clustering elements in the dataset $\mathcal{A}$ into n groups.

Algorithm 2 Path Set Update: PSU($\mathcal{L}, \mathcal{G}_3^1$)

```
Input path set $\mathcal{L}$, node set $\mathcal{G}_3^1$, $i = 1$;
while $i \leqslant L$ do
    if $\mathcal{G}_3^1 \cap \mathfrak{l}_i \neq \varnothing$ then
        $\mathcal{L} \leftarrow \mathcal{L} \setminus \mathfrak{l}_i$;
    end if
    $i \leftarrow i + 1$;
end while
Output updated $\mathcal{L}$.
```

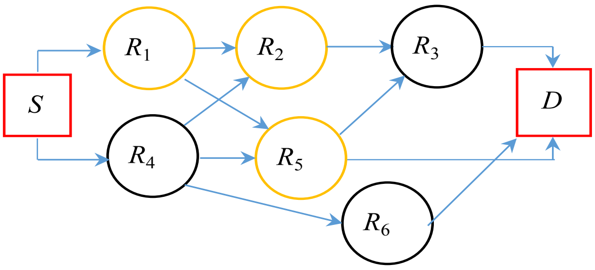


Figure 3: An example for ESD Algorithm

estimated contribution values at an intermediate level. The reason is that a node's contribution is influenced by the behavior of other nodes along its associated multihop path(s). In K-means clustering with K=2, these intermediate levels will still be assigned to either benign or malicious, resulting in misdetection (malicious nodes identified as benign) and false alarms (benign nodes identified as malicious).

To solve this problem, we propose soft detection, where we cluster nodes into three groups, instead of two. The three clusters represent high, medium and low contribution values. Here, we identify the highly suspicious group (nodes with high P_i) as malicious first and then identify other nodes. The corresponding node sets to the high, medium, low contribution values are denoted as $\mathcal{G}_1^1$, $\mathcal{G}_2^1$ and $\mathcal{G}_3^1$, respectively. All nodes in $\mathcal{G}_3^1$ are identified as malicious. However, we do not directly identify nodes in $\mathcal{G}_1^1$ as benign and $\mathcal{G}_2^1$ as malicious (with low P_i) since their contribution values can be calculated very inaccurately when nodes in $\mathcal{G}_3^1$ are involved. Hence, we try to refine the feature (contribution value) calculations by removing nodes in $\mathcal{G}_3^1$ from the computations whenever possible (this is done by not including their related paths when calculating contribution values).

More specifically, our approach first records the original path set $\mathcal{L}_0$ and contribution set $\mathcal{C}_0$ as $\mathcal{L}_0 = \mathcal{L}$, $\mathcal{C}_0 = \mathcal{C}$. Then, it excludes from the path set $\mathcal{L}$ any path that contains any of the nodes in $\mathcal{G}_3^1$. That way, $\mathcal{L}$ is updated. The steps to launch the path set update are elaborated in Algorithm 2 (Path Set Update (PSU)). After updating the path set $\mathcal{L}$, we recalculate the contribution set $\mathcal{C}$ for the remaining (unidentified) node set $\mathcal{G}_U \triangleq \mathcal{G}_1^1 \cup \mathcal{G}_2^1$. Then, we apply K-means($\mathcal{C}$,2) to cluster nodes in $\mathcal{G}_U$ into two sets: $\mathcal{G}_1^2$ and $\mathcal{G}_2^2$. The nodes with higher contribution values ($\mathcal{G}_1^2$) are classified as benign, while others ($\mathcal{G}_2^2$) are classified as malicious. We describe the soft detection approach in Algorithm 3.

Condition 1: Our approach for malicious node identification depends on sufficient network diversity to calculate accurate nodes' contribution values (trustworthiness). After path set update, nodes in $\mathcal{G}_U$ along with S and D form another graph topology different from the original topology formed by all the nodes. We define $\tilde{d}_U$ as the average node degree of the graph formed by nodes in $\mathcal{G}_U$, S and D. If $\tilde{d}_U$ is lower than a predefined threshold η, then the diversity in the remaining topology is insufficient to evaluate new contribution values. In this case, we apply the HD algorithm to identify the behavior of nodes in $\mathcal{G}_U$. We set $\eta > 2$ to ensure that every node is connected to more than one path. For example, for the network in Fig. 2, if $\mathcal{G}_3^1$=$\{R_1, R_2\}$, then the updated $\mathcal{L}$ contains two paths only, i.e., R_4-R_5-R_3 and R_4-R_5, to form a graph (that includes S and D). After discarding $\mathcal{G}_3^1$, the average node degree is $\tilde{d}_U$=2. This expresses that an intermediate node is connected to one path only (has a link with a preceding node along a path and another link with a succeeding node, hence its degree is 2). Since $\tilde{d}_U \leqslant \eta$, SD cannot be applied and HD is applied instead. In some cases, the updated $\mathcal{L}$ may even contain no paths after nodes in $\mathcal{G}_3^1$ are removed from the original network topology; in these cases, we also apply HD results to nodes in $\mathcal{G}_U$.

Condition 2: Even if $\tilde{d}_U > \eta$, there may be some nodes in $\mathcal{G}_U$ not associated with any path in the updated $\mathcal{L}$. These nodes still remain unidentified after executing $(\mathcal{G}_1^2, \mathcal{G}_2^2) = \text{HD}(\mathcal{C}, \varepsilon, \mathcal{L})$ (Line 14,15 in SD Algorithm, i.e., Algorithm 3). In this case, the identification results from HD (using the original topology as the input) are applied to identify the behavior of these nodes.

5.2 Enhanced Soft Detection (ESD)

We propose an enhanced soft detection (ESD) algorithm, which improves the detection accuracy of the SD algorithm. In particular, we here utilize benign paths (all nodes along them are benign) to correct misdetection or false alarms of the SD algorithm. From our previous discussions, one can infer the following: (i) if a path's reputation is 1, then nodes along this path do not manipulate packets. In other words, all nodes along this path are benign. Nodes satisfying such conditions are classified into the benign node set $\mathcal{G}_B^{(ESD)}$; (ii) for a path

Algorithm 3 Soft Detection: SD($\mathcal{C}, \varepsilon, \eta, \mathcal{L}$)

```
Input dataset $\mathcal{C}$, threshold $\varepsilon$, $\eta$, path set $\mathcal{L}$;
if $\mathcal{C} = \bar{1}$ then
    $\mathcal{G}_B^{(SD)} = \{1, ..., N\}, \mathcal{G}_M^{(SD)} = \varnothing$;
else if $\tilde{T} < \varepsilon_1$ then
    $\mathcal{G}_M^{(SD)} = \{1, ..., N\}, \mathcal{G}_B^{(SD)} = \varnothing$;
else
    Use k-means($\mathcal{C}$,3) to cluster $\mathcal{C}$ into three groups where the group with the lowest data values is the malicious node set $\mathcal{G}_3^1$; the other two groups $\mathcal{G}_1^1$ and $\mathcal{G}_2^1$ are considered as an unidentified node set $\mathcal{G}_U \triangleq \mathcal{G}_1^1 \cup \mathcal{G}_2^1$;
    $\mathcal{L}_0 = \mathcal{L}$; $\mathcal{C}_0 = \mathcal{C}$; $(\mathcal{G}_B^0, \mathcal{G}_M^0) = $ HD$(\mathcal{C}_0, \varepsilon, \mathcal{L}_0)$;
    Update $\mathcal{L} = $ PSU$(\mathcal{L}, \mathcal{G}_3^1)$; Update $\mathcal{C}, \mathcal{T}$ based on $\mathcal{L}$;
    if ($\tilde{d_U} \leqslant \eta$ and $\mathcal{C} \neq \bar{1}$) or $|\mathcal{L}| = 0$ then
        Apply the results from $\mathcal{G}_B^0, \mathcal{G}_M^0$ to nodes in $\mathcal{G}_U$;
        Let $\mathcal{G}_M^{(SD)}$ be the union of $\mathcal{G}_3^1$ and malicious nodes in $\mathcal{G}_U$, and the remaining nodes belong to $\mathcal{G}_B^{(SD)}$;
    else
        $(\mathcal{G}_1^2, \mathcal{G}_2^2)$=HD$(\mathcal{C}, \varepsilon, \mathcal{L})$;
        $\mathcal{G}_M^{(SD)}$=$\mathcal{G}_3^1 \cup \mathcal{G}_2^2$; $\mathcal{G}_B^{(SD)}$=$\mathcal{G}_1^2$;
    end if
end if
Output benign node set: $\mathcal{G}_B^{(SD)}$; malicious node set: $\mathcal{G}_M^{(SD)}$.
```

with reputation less than 1, if there is only one node in this path not belonging to $\mathcal{G}_B^{(ESD)}$, then that node is malicious since it is the only possible node to manipulate packets. Nodes satisfying such conditions are classified into malicious node set $\mathcal{G}_M^{(ESD)}$. For example, consider the topology in Fig. 3, where R_3, R_4 and R_6 are benign and R_1, R_2 and R_5 are malicious. D can observe that the reputation of path "$R_4 - R_6$" is 1 and accordingly infer that both R_4 and R_6 are benign. Hence, $\mathcal{G}_B^{(ESD)} = \{R_4, R_6\}$. Then , D can also observe that the reputation of path "$R_4 - R_5$" is less than 1. Since R_5 is the only node in path "$R_4 - R_5$" not belonging to $\mathcal{G}_B^{(ESD)}$, R_5 is identified as malicious and put into $\mathcal{G}_M^{(ESD)}$. There are no other paths satisfying inference (ii), so we identify R_1, R_2, R_3 using the results from SD Algorithm.

ESD Algorithm is executed after SD to improve the detection accuracy. That is, if there is any identification result conflict from SD and ESD to R_i, we accept the result from ESD for R_i. Note that ESD algorithm can also be used to improve the results from HD algorithm in a similar way. The ESD algorithm is described in Algorithm 4.

Algorithm 4 Enhanced Soft Detection: ESD($\mathcal{L}_0$)

```
Define $\mathcal{U}$ as the set containing all the nodes;
Input path set $\mathcal{L}_0$, node set $\mathcal{G}_B^{(ESD)} = \mathcal{G}_M^{(ESD)} = \varnothing$, $i = 1$;
while $i \leqslant L$ do
    if $T_{\mathfrak{l}_i} = 1$ then
        $\mathcal{G}_B^{(ESD)} \leftarrow \mathcal{G}_B^{(ESD)} \cup \mathfrak{l}_i$;
    end if
    $i \leftarrow i + 1$;
end while
$(\mathcal{G}_B^{(ESD)})^C \leftarrow \mathcal{U} - \mathcal{G}_B^{(ESD)}$, $i = 1$;
while $i \leqslant L$ do
    if $T_{\mathfrak{l}_i} \neq 1$ and $|\mathfrak{l}_i \cap (\mathcal{G}_B^{(ESD)})^C| = 1$ then
        $\mathcal{G}_M^{(ESD)} \leftarrow \mathcal{G}_M^{(ESD)} \cup (\mathfrak{l}_i \cap (\mathcal{G}_B^{(ESD)})^C)$;
    end if
    $i \leftarrow i + 1$;
end while
Output benign node set: $\mathcal{G}_B^{(ESD)}$ and malicious node set: $\mathcal{G}_M^{(ESD)}$.
```

Algorithm 5 Reputation Correction (RC)

```
Input path set $\mathcal{L}$, path reputation set $T$, $i = 1$;
while $i \leqslant L$ do
    $T_{\mathfrak{l}_i} \leftarrow T_{\mathfrak{l}_i} / (\bar{\mu}^{(|\mathfrak{l}_i|+1)})$;
    if $T_{\mathfrak{l}_i} > 1$ then
        $T_{\mathfrak{l}_i} \leftarrow 1$;
    end if
    $i \leftarrow i + 1$;
end while
```

5.3 Reputation Correction under Channel Errors

In this subsection, we discuss the impact of channel errors on the detection algorithms. We consider that the channel packet error rate is the same for all communication links, and is denoted by μ. Thus, the rate of correct transmission per link is $\bar{\mu} = 1 - \mu$. μ can be estimated by pilot or training packets. We can follow a similar analysis in Section 4.1 to obtain nodes' contribution values. For example, for the link $S - R_1$, where S sends Q packets to R_1, the expected number of correctly received packets is $Q\bar{\mu}$. Then, consider a path of n hops, the expected path reputation is $\bar{\mu}^n \prod_{i=1}^{n-1} \bar{P}_i$. To cancel the effect of channel errors from the feature computations, we divide by $\bar{\mu}^n$. For example, the effect of channel errors on $T_{1,2,3}$ can be canceled by updating: $T_{1,2,3} \leftarrow T_{1,2,3}/(\bar{\mu}^4)$. Then all the updated reputation values are utilized to calculate the nodes' contributions and then identify malicious nodes by the SD Algorithm in a similar way.

The above steps are illustrated in Algorithm 5. It is also noted that if we input the corrected reputations $\mathcal{T}$ into SD Algorithm (Algorithm 3), some minor modifications are needed[4]. More specifically, "$\mathcal{C} = \bar{1}$" in Line 2 should be changed to "avg$(\mathcal{C}) \geqslant \varepsilon_{RC}$" and "$\mathcal{C} \neq \bar{1}$" in Line 10 should be changed to "avg$(\mathcal{C}) < \varepsilon_{RC}$" to tolerate the channel error effect. ε_{RC} is very close to 1 and "avg$(\mathcal{C})$" refers to the average value of elements in set $\mathcal{C}$. Note that $\bar{\mu}^n < 1$. Due to possible estimation errors, after updating $T_{\mathfrak{l}_i}$ by canceling channel error effects we may get $T_{\mathfrak{l}_i} > 1$, which is not reasonable. In this case, we set $T_{\mathfrak{l}_i} = 1$.

6 Interactions among Small-Scale Networks via LoRa

6.1 Information via LoRa Links

Recall our proposed network division mechanism in Section 3. $\mathfrak{R}_k$ is a trusted DL node, which acts as the destination of probe packets for Region $\mathfrak{R}_j - \mathfrak{R}_k$. According to Algorithm 1, 3, 4, $\mathfrak{R}_k$ should check all packets' integrity, calculate $\mathcal{T}, \mathcal{C}$ and apply K-means machine learning method for clustering nodes' behavior based on contribution values. However, in case $\mathfrak{R}_k$ does not have sufficient computational resources, some of these functions can be performed at the LoRa gateway.

As explained in Section 3, each DL node has two communication interfaces, one of which is the long-range LoRa link interface. Each DL node can communicate with a distant LoRa gateway directly; the LoRa gateway typically has high computational power, and we assume it knows the network topology. There are two possible cases for identification of malicious nodes:

- *Case 1:* the DL nodes have sufficient computational power. In this case, each DL node computes the nodes' contribution values and executes K-means to obtain the identification results for its corresponding network region. Then, each DL node delivers the identification results via LoRa links to the LoRa gateway.
- *Case 2:* the DL nodes do not have sufficient computational power. Hence, a DL node delivers the reputation value set $\mathcal{T}$ (which is a short message) via LoRa links to the LoRa gateway. The gateway knows the network topology, and it can compute the contribution set $\mathcal{C}$ and apply K-means clustering to get the classification results.

Lastly, via LoRa links, the LoRa gateway notifies each DL node to eliminate the malicious nodes from the routing paths. This notification can also be transmitted by DL nodes through reliable paths of short-range links.

Remark 1: Some SL nodes can be involved in multiple regions simultaneously. For example, in Fig. 4, an SL node, denoted as R_s, is involved in Region $\mathfrak{R}_4 - \mathfrak{R}_5$ and $\mathfrak{R}_3 - \mathfrak{R}_5$. These two regions can have different identification results for R_s. Our approach is to accept the identification result of R_s from the region with the highest average node degree. The reason is that higher average node degree leads to higher diversity of paths in a region, which helps in classifying R_s's behavior more accurately.

6.2 Path Selection

After malicious node identification, reliable multihop paths (of short-range links) can be selected for transmissions. Whenever a DL node receives a packet, it attempts to forward it over the most reliable path. When the packet source and destination are in distant regions, DL nodes can select which (intermediate) regions a packet will go through until it reaches its destination. Here, we propose a strategy for path selection in the large-scale network, where we consider each small-scale region $\mathfrak{R}_j - \mathfrak{R}_k$ as a "virtual node"; the highest path reputation within this region is assigned to virtual node as the virtual node contribution. Each DL node is also considered as a virtual node; its virtual node contribution value is 1 (since it is assumed to be trusted). As shown in Fig. 5, these virtual nodes constitute various paths, named "virtual paths". For example, a virtual path can be constituted by: $\mathfrak{R}_1$, $\mathfrak{R}_1 - \mathfrak{R}_3$, $\mathfrak{R}_3$, $\mathfrak{R}_3 - \mathfrak{R}_5$, $\mathfrak{R}_5$. From the analysis in Section 4.1, each virtual path in Fig. 5 also has virtual path reputation, which is obtained by multiplying the contribution values of virtual nodes along this virtual path. In this way, virtual nodes along the virtual path with the highest reputation can be selected as the most reliable path for packet forwarding.

7 Numerical Results

In this section, we demonstrate the effectiveness of the architecture and evaluate the performance of the malicious nodes identification techniques (HD and SD). The ESD Algorithm is applied to both HD and SD algorithms. We compare our methods with the conventional watchdog scheme. We illustrate the effect of the maximum number of hops, network diversity, percentage of malicious nodes (in each small-scale network) and the number of DL nodes (in a large-scale network) on the detection accuracy. We show the impact of the architecture that utilizes dual link technologies (LoRa and short range RF) in improving the network security by accurately identifying malicious nodes.

[4] Note that ESD Algorithm is not applicable to situations under channel errors since inference (i) in Section 4 would be incorrect with channel errors.

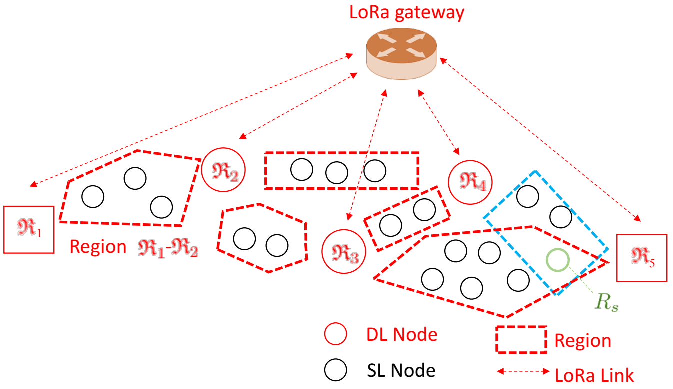


Figure 4: Large-scale network with one SL node involved in two Regions

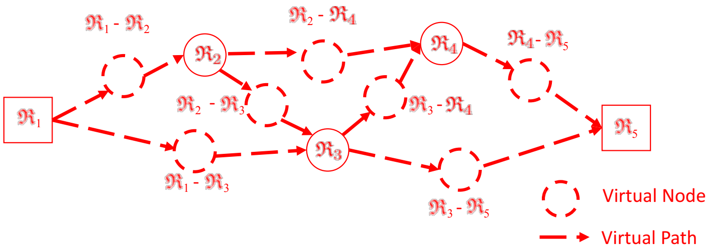


Figure 5: Large-scale network with DL nodes in a virtual view

7.1 Simulation Setup and Performance in a Single Region

In the simulation for small-scale networks, we assume there are N single-link (SL) relay nodes uniformly distributed in a $(6N)\times15\text{m}^2$ rectangular network region. The SL node density is 0.01/m^2. The DL nodes S and D are located at the midpoint of the left and right edges of the rectangle. The communication range over the short-range RF link is r=29m. We generate 20 random networks. For each network, simulations are repeated in 500 rounds. In each round, unless stated otherwise, we randomly choose 30% of the nodes to be malicious. Each malicious node's attack probability P_i is a random value in the range of [0.2,0.8] where values are uniformly distributed; once assigned, P_i is fixed throughout a round of simulation. We obtain all possible paths from S and D and randomly select some paths. Each node forwards a packet to the next hop that is closer to D. We utilize $Q = 200$ probe packets transmitted through each path[5]. From experimental observations, we set the thresholds $\eta = 3.7$ and $\varepsilon = 0.0009$.

To evaluate the proposed approaches, we evaluate the detection accuracy and false alarm rate. The detection accuracy is defined as: P_d = Number of correctly identified malicious nodes / Number of malicious nodes. The false alarm rate is defined as: F_a =Number of benign nodes identified as malicious / Number of benign nodes. All the results are measured and averaged based on all simulation rounds over 20 random networks, which are executed in Matlab.

Example 1: impact of the maximum number of hops. In this example, we show the impact of the maximum number of hops on P_d and F_a of HD and SD algorithms with N=10 and 33 paths. Since each path may have different number of hops, we denote the maximum number of hops along a path as N_p; that is, each path selected for packet transmission has no more than N_p hops. P_d and F_a of HD and SD versus N_p are plotted in Fig. 6. One can observe that larger N_p causes P_d to decrease and F_a to increase. The reason is that, when the percentage of malicious nodes is fixed, more hops introduce higher uncertainty (the unknown P_i of each malicious node) to the network. Hence, the number of hops along a path should be limited.

Example 2: impact of network diversity. Here, we examine the impact of network diversity on detection accuracy with N=10. We limit the number of hops to N_p=6. Moreover, based on the selected 33 paths, we randomly choose 20%, 40%, 60%, 80%, 100% of these 33 paths to be used for packet transmission, as shown in Fig. 7. It can be demonstrated that when more paths are used, the performance improves (P_d increases and F_a

[5] S transmits 200 packets in one path and then switches to another path to start the transmission.

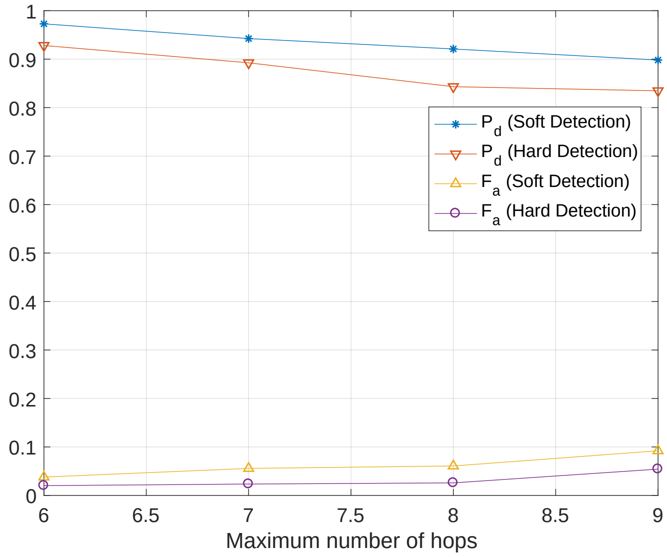


Figure 6: P_d and F_a as functions of maximum number of hops with 30% malicious nodes and 33 paths

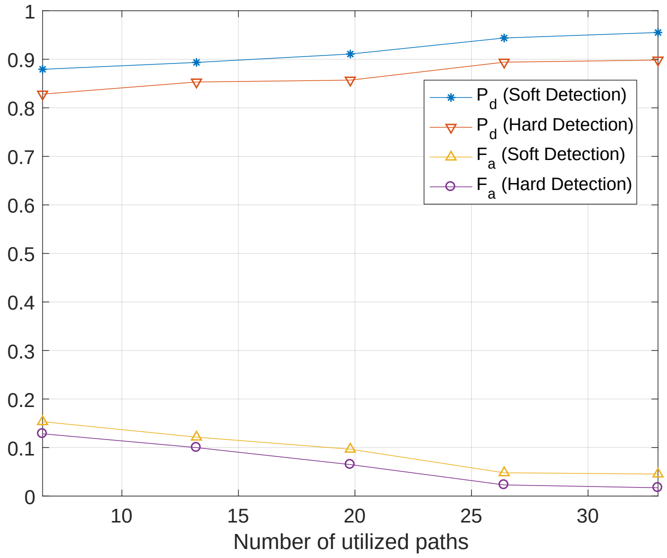


Figure 7: P_d and F_a as functions of number of paths with N=10, N_p=6, 30% malicious nodes

decreases). The figure shows the performance of both HD and SD. It can be seen that SD has high detection accuracy than HD, at the expense of slightly higher false alarm rate.

Example 3: Impact of percentage of malicious nodes. In this example, we set N=10, N_p=6 and examine the impact of the percentages of the malicious nodes on the performance of approaches. In Fig. 8, we plot P_d and F_a as functions of percentage of malicious nodes. As expected, P_d decreases as the percentage of malicious nodes increases. This is because there are fewer reliable paths. We also examine the performance of SD Algorithm in the presence of channel errors, where $\mu = 2\%$. By applying the reputation correction described earlier, Fig. 8 demonstrates that the detection accuracy of SD under channel errors is close to the case without channel errors.

We also compare our algorithms with the conventional watchdog scheme [7], which is a well-known technique used for detecting packet manipulation attacks. In the watchdog scheme, every node overhears packets forwarded by neighboring nodes, and verifies whether these packets were manipulated or not. Then, each node reports its opinion about the trustworthiness of

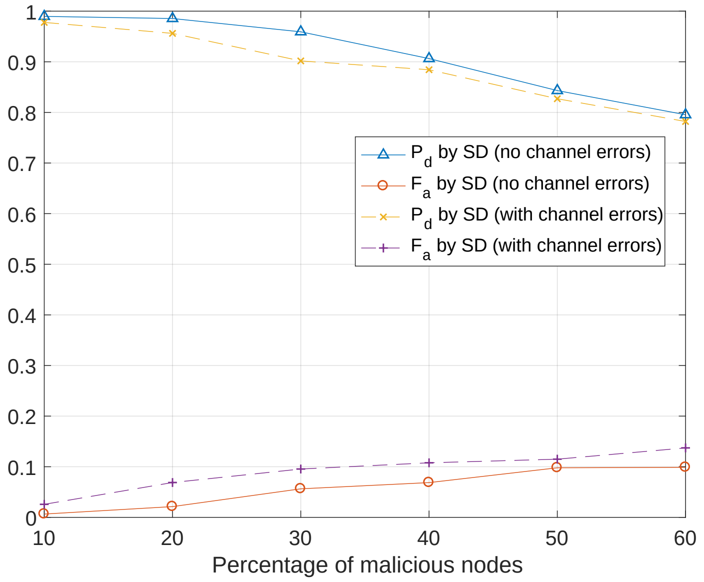


Figure 8: P_d **and** F_a **as functions of percentage of malicious nodes with** N**=10,** N_p**=6, 33 paths and** μ**=2%**

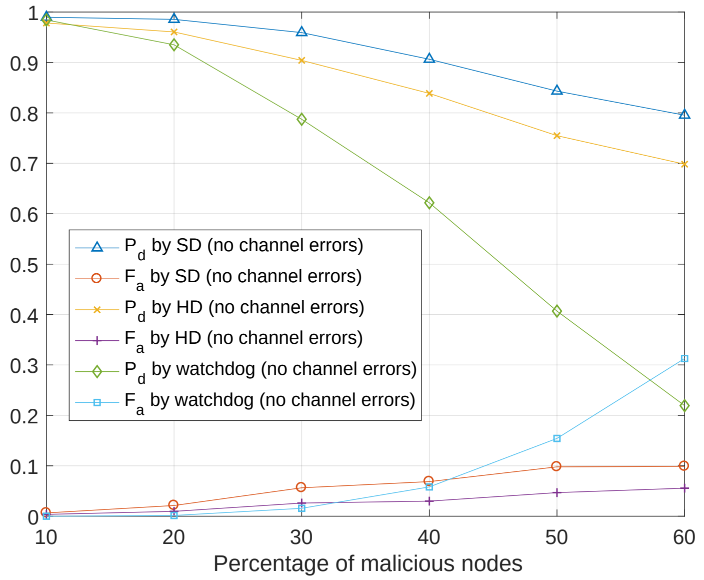


Figure 9: P_d **and** F_a **as functions of percentage of malicious nodes with** N**=10,** N_p**=6, and 33 paths**

neighboring nodes to the destination. By collecting reports from all nodes, the destination applies majority voting to identify whether a node is malicious or not. We use the same attack model described in Section 3.3 for packet manipulation. Furthermore, compromised nodes can report falsified information about their neighbors, i.e., report malicious nodes as benign and benign nodes as malicious, which is what we assumed here. It is observed from Fig. 9 that, when the percentage of malicious nodes is between 20% to 40%, our algorithms achieve higher detection accuracy than the watchdog scheme with slightly higher false alarm rate. In addition, the false alarm rate of the watchdog scheme is higher than that of our algorithms when the percentage of malicious nodes exceeds 40%. It is worth mentioning that the watchdog scheme would require extra energy, computational and memory resources at each node to overhear all their neighbors' packets and evaluate their trustworthiness. The computational burden in our proposed approaches is moved from the simple SL nodes to the more powerful DL nodes and/or LoRa gateways.

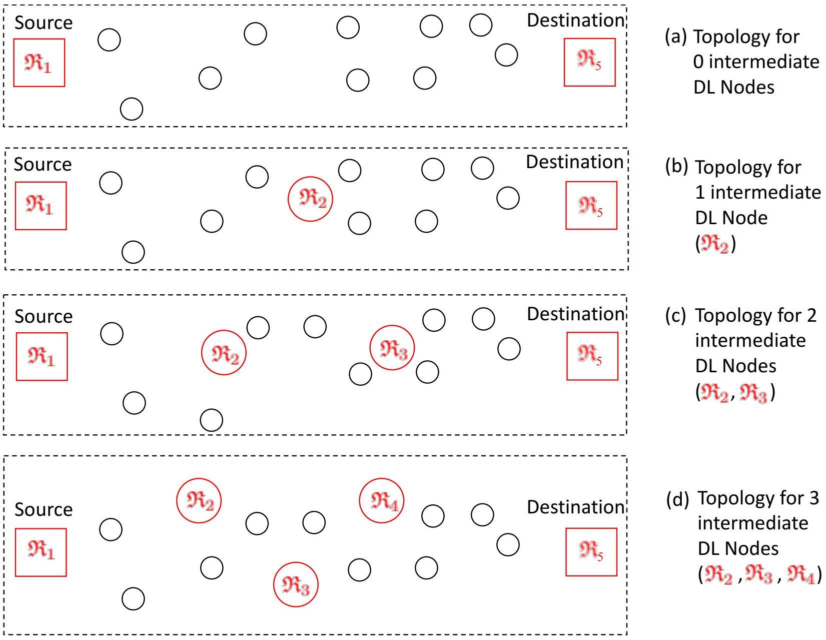


Figure 10: Four types of topologies with 0, 1, 2 or 3 intermediate DL nodes

7.2 Performance Gains using Long-Range Links

Consider a rectangular network area of $(6N)\times 15\text{m}^2$, with $N = 30$ SL nodes uniformly deployed (SL node density is still $0.01/\text{m}^2$) and DL nodes $\mathfrak{R}_1$ and $\mathfrak{R}_5$ at the edge as shown in Fig. 10 (a)(b)(c)(d). We examine the performance when there are none, one, two, or three additional DL nodes in the area. That is, we consider the scenarios where the whole network is treated as a single region, or divided into two, three or seven[6] regions of smaller scales. Each neighboring DL node acts as source/destination of probe packets in their corresponding region(s). The source in each region uses multiple paths to transmit packets to the destination in the same region. In our experiments, we set the locations of intermediate DL nodes such that the smaller-scale regions are of almost the same size. In general, DL nodes can be uniformly distributed. The LoRa RF link communication range with low packet loss can be up to 5km [25]. That is, in each topology of Fig. 10, assuming that the LoRa gateway is placed within 5km to each DL node, DL nodes can communicate with each other using LoRa RF links. Other settings are the same as the previous subsection [7].

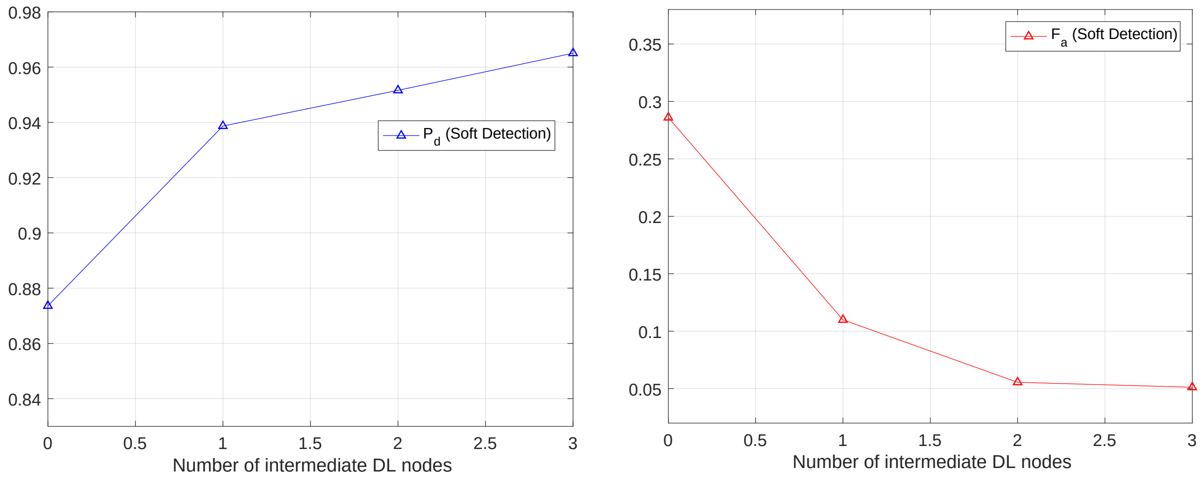
Example 4: impact of number of DL nodes. We show the relationship between the detection accuracy and the number of intermediate DL nodes in Fig. 11. It is observed that the detection accuracy grows and false alarm reduces when the number of intermediate DL nodes increases. That is, utilizing more dual links in the network architecture enables more accurate identification of malicious nodes, and requires much lower overhead compared to our previously proposed methods, such as in [15][16].

8 Conclusions and Future Work

In this paper, we proposed a novel hierarchical network architecture design that utilizes dual communication link technologies with distinct characteristics to facilitate inference about the trustworthiness of network nodes. The proposed network architecture is hierarchical, where trusted DL nodes divide the large-scale network into several small-scale network portions that are more reliable and easier to manage. For each network portion, we proposed to use unsupervised learning that exploits the diversity of network paths to identify malicious nodes launching packet manipulation attacks in a multihop IoT network. We formulated nodes' trustworthiness metrics to cluster nodes according to their behavior. Two algorithms were proposed: Hard Detection (HD) and Soft Detection (SD). The HD algorithm clusters nodes into benign and malicious groups. To further consider the variability of attack probabilities, the SD algorithm clusters nodes into three groups based on

[6] The seven regions in Fig. 10 (d) are shown clearly in Fig. 5.

[7] To be consistent with to the previous subsection, the ratio between the number of paths and the number of nodes per region is used here. Hence, we set number of used paths to around 3.3 times the number of SL nodes in the same region.



(a) Detection accuracy

(b) False alarm rate

Figure 11: P_d and F_a as functions of number of intermediate DL nodes with 30 SL nodes, 30% malicious SL nodes

their suspicious levels; then highly suspicious nodes are removed and more accurate trustworthiness metrics (contribution values) are calculated for the remaining nodes provided sufficiently high network diversity. We also analyzed the impact of channel errors on the detection performance. Simulation results showed that our malicious node identification techniques have higher detection accuracy compared to the conventional watchdog scheme. The gains increase as the percentage of malicious nodes increases. It is observed that increasing the density of DL nodes further improves the detection accuracy since it enables more accurate trustworthiness evaluation of the meshed SL nodes.

In this paper we assumed that DL nodes are reliable and trusted. The problem becomes challenging if some DL nodes are compromised. Also, in our approaches, we randomly selected some routes from all possible paths to transmit packets; choosing only few paths with specific features could reduce the transmission overhead while maintaining the detection accuracy. We are exploring these issues as part of our ongoing work.

References

[1] C. Withanage, R. Ashok, C. Yuen, and K. Otto, “A comparison of the popular home automation technologies,” in *Innovative Smart Grid Technologies-Asia (ISGT Asia)*. IEEE, 2014, pp. 600–605.

[2] J. Zheng and M. J. Lee, “A comprehensive performance study of ieee 802.15. 4,” *Sensor network operations*, vol. 4, pp. 218–237, 2006.

[3] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, “Ddos in the iot: Mirai and other botnets,” *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[4] S. Kriaa, M. Bouissou, and L. Piètre-Cambacédès, “Modeling the stuxnet attack with bdmp: Towards more formal risk assessments,” in *7th International Conference on Risk and Security of Internet and Systems (CRiSIS)*. IEEE, 2012, pp. 1–8.

[5] S. J. Shackelford, A. A. Proia, B. Martell, and A. N. Craig, “Toward a global cybersecurity standard of care: Exploring the implications of the 2014 nist cybersecurity framework on shaping reasonable national and international cybersecurity practices,” *Tex. Int'l LJ*, vol. 50, p. 305, 2015.

[6] C. Wang, T. Feng, J. Kim, G. Wang, and W. Zhang, “Catching packet droppers and modifiers in wireless sensor networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 5, pp. 835–843, 2012.

[7] K. Ioannis, T. Dimitriou, and F. C. Freiling, “Towards intrusion detection in wireless sensor networks,” in *Proc. of the 13th European Wireless Conference*, 2007, pp. 1–10.

[8] M. Rizzi, P. Ferrari, A. Flammini, and E. Sisinni, “Evaluation of the iot lorawan solution for distributed measurement applications,” *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 12, pp. 3340–3349, 2017.

[9] X. Liu, M. Abdelhakim, P. Krishnamurthy, and D. Tipper, “Identifying malicious nodes

in multihop iot networks using diversity and unsupervised learning," in *IEEE International Conference on Communications*. IEEE, 2018, pp. 1–6.

[10] R. Akbani, T. Korkmaz, and G. Raju, "A machine learning based reputation system for defending against malicious node behavior," in *Global Telecommunications Conference*. IEEE, 2008, pp. 1–5.

[11] J. Dromard, G. Roudiere, and P. Owezarski, "Online and scalable unsupervised network anomaly detection method," *IEEE Transactions on Network and Service Management*, vol. 14, no. 1, pp. 34–47, 2017.

[12] X. Yang, J. Lin, W. Yu, P.-M. Moulema, X. Fu, and W. Zhao, "A novel en-route filtering scheme against false data injection attacks in cyber-physical networked systems," *IEEE Transactions on Computers*, vol. 64, no. 1, pp. 4–18, 2015.

[13] M. Abdelhakim, L. E. Lightfoot, J. Ren, and T. Li, "Distributed detection in mobile access wireless sensor networks under byzantine attacks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 4, pp. 950–959, 2014.

[14] T. Hayajneh, P. Krishnamurthy, D. Tipper, and T. Kim, "Detecting malicious packet dropping in the presence of collisions and channel errors in wireless ad hoc networks," in *IEEE International Conference on Communications*. IEEE, 2009, pp. 1–6.

[15] M. Abdelhakim, L. Lightfoot, J. Ren, and T. Li, "Reliable communications over multihop networks under routing attacks," in *Global Communications Conference (GLOBECOM)*. IEEE, 2015, pp. 1–6.

[16] M. Abdelhakim, X. Liu, and P. Krishnamurthy, "Diversity for detecting routing attacks in multihop networks," in *International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2018, pp. 1–6.

[17] S. Kaplantzis, A. Shilton, N. Mani, and Y. A. Sekercioglu, "Detecting selective forwarding attacks in wireless sensor networks using support vector machines," in *3rd International Conference on Intelligent Sensors, Sensor Networks and Information*. IEEE, 2007, pp. 335–340.

[18] J. F. C. Joseph, B.-S. Lee, A. Das, and B.-C. Seet, "Cross-layer detection of sinking behavior in wireless ad hoc networks using svm and fda," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 2, pp. 233–245, 2011.

[19] A. I. Moustapha and R. R. Selmic, "Wireless sensor network modeling using modified recurrent neural networks: Application to fault detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 5, pp. 981–988, 2008.

[20] D. Janakiram, V. Reddy, and A. P. Kumar, "Outlier detection in wireless sensor networks using bayesian belief networks," in *First International Conference on Communication System Software and Middleware*. IEEE, 2006, pp. 1–6.

[21] K. Leung and C. Leckie, "Unsupervised anomaly detection in network intrusion detection using clusters," in *Proceedings of the Twenty-eighth Australasian conference on Computer Science*, vol. 38. Australian Computer Society, Inc., 2005, pp. 333–342.

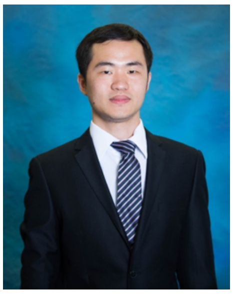
[22] S. Ruoti, S. Heidbrink, M. O'Neill, E. Gustafson, and Y. R. Choe, "Intrusion detection with unsupervised heterogeneous ensembles using cluster-based normalization," in *IEEE International Conference on Web Services*. IEEE, 2017, pp. 862–865.

[23] K. Nahiyan, S. Kaiser, K. Ferens, and R. McLeod, "A multi-agent based cognitive approach to unsupervised feature extraction and classification for network intrusion detection," in *International Conference on Advances on Applied Cognitive Computing (ACC)*. CSREA, 2017, pp. 25–30.

[24] K. J. Cios, W. Pedrycz, R. W. Swiniarski, and L. A. Kurgan, *Data mining: a knowledge discovery approach*. Springer Science & Business Media, 2007.

[25] J. Petäjäjärvi, K. Mikhaylov, M. Pettissalo, J. Janhunen, and J. Iinatti, "Performance of a low-power wide-area network based on lora technology: Doppler robustness, scalability, and coverage," *International Journal of Distributed Sensor Networks*, vol. 13, no. 3, p. 1550147717699412, 2017.

AUTHOR BIOGRAPHIES



Xin Liu received the B.E. degree from the School of Electronic Information, Wuhan University, China, in 2014, and the M.S. degree from the Department of Electrical and Computer Engineering, Western University, Canada, in 2016. He is currently pursuing the Ph.D. degree with the School of Computing and Information, University of Pittsburgh, USA. His research interests include machine learning, Internet of Things and urban computing.

Mai Abdelhakim is a visiting assistant professor in the School of Computing and Information at the University of Pittsburgh. Prior to joining the University of Pittsburgh in 2016, she was a postdoctoral research scientist at OSRAM Research Center (2015-2016). In 2014, she was a postdoctoral research associate at Michigan State University. She also worked at the German University in Cairo (2007-2008) and at the Egyptian National Center for Radiation Research and Technology (2008-2010). She received her Ph.D. degree in Electrical Engineering from Michigan State University in 2014, and Bachelor's and Master's degrees in Electronics and Communications Engineering from Cairo University in 2006 and 2009, respectively. Her research interests include reliable and secure cyber-physical systems, network design, signal processing and statistical learning.

Mai Abdelhakim is the corresponding author of this work.

Prashant Krishnamurthy is a professor in the School of Computing and Information at the University of Pittsburgh. He is a co-founder of the Laboratory of Education and Research in Security Assured Information Systems (LERSAIS), which has been designated as Center of Academic Excellence in information Education and Research (CAE + CAE-R) jointly by the United States NSA and DHS. His research interests include Wireless Network Security, Positioning and Localization, and Cryptography and Information Security. He has had research funding from the National Science Foundation, the National Institute of Standards and Technology, and The Army Research Office.

David Tipper is Professor in the School of Computing and Information at the University of Pittsburgh. Dr. Tipper is a graduate of the University of Arizona (Ph.D. E.E., M.S. S.I.E.) and Virginia Tech (B.S.E.E). At Pitt he served as Director of the Graduate Telecommunications and Networking Program from 2007 – 2016. His current research interests are wired and wireless network design, performance analysis, network security and critical infrastructure resilience. His teaching and research has been supported by grants from various government and corporate sources such as NSF, ARO, NIST, NSA, IBM, Bechtel Bettis and AT&T. He is a Senior member of the Institute of Electrical and Electronics Engineers (IEEE) and has been on over 30 international conference technical committees. Further, he has co-edited five special issues of journals including two feature issues on Advances in Network Planning for IEEE Communications (Jan., Feb., 2014). Additionally, he has co-authored two books including one focused on reliability and cybersecurity and over 150 research articles in books, journals, and refereed conference proceedings.