Masih Abedini, candidate for the degree of Master of Applied Science in Electronic Systems Engineering, has presented a thesis titled, *Active eavesdroppers detection system in multi-hop wireless sensor networks*, in an oral examination held on July 25, 2022. The following committee members have found the thesis acceptable in form and content, and that the candidate demonstrated satisfactory knowledge of the subject material.

External Examiner: Dr Nashid Shahriar, Department of Computer Science*

Supervisor(s): Dr Irfan Al-Anbagi, Department of Electronic Systems Engineering*

Committee Member: Dr Paul Laforge, Department of Electronic Systems Engineering *

Chair of Defense: Dr Andrei Volodin, Department of Mathematics and Statistics*

*Attended via Zoom conferencing
Abstract

Wireless Sensor Networks (WSNs) are vulnerable to eavesdropping attacks that endanger their privacy, confidentiality, and authenticity. As the broadcast nature of the wireless channel makes it susceptible to eavesdropping by adversaries, the detection of eavesdroppers in wireless networks can lessen the chances of more damaging attacks. Historically, researchers have attempted to reduce the risk of covert eavesdropping through the use of cryptographic protocols, information-theoretic solutions, and transmission range control. These methods are not suitable for WSNs with resource constraints. It is noteworthy that active eavesdroppers are legitimate nodes that are compromised by adversaries to eavesdrop on traffic while performing their normal responsibilities in ad-hoc networks. Detecting such malicious nodes slows the ongoing destructive attacks.

In this thesis, we present a novel Active Eavesdroppers Detection (AED) system for homogeneous multi-hop WSNs. The AED system consists of two major modules: a Monitoring module and a Detection Engine module. The Monitoring module plays a vital role in the AED system to provide accurate measurements for the Detection Engine module. The Detection Engine module is provided with a lightweight de-tection engine module that employs the Z-test method and runs on edge devices.

Regarding measurements, we first use intra-node delay measurements as the
input feature of the AED system. To measure intra-node delays of nodes, the Monitor-ing module employs an out-of-band monitoring system using static nodes, Unmanned Aerial Vehicles (UAVs), or both of them. According to simulation results in the Cooja and MATLAB environments, the AED system can detect active eavesdroppers who relay packets to their neighbors. However, it fails to detect active eavesdroppers who do not forward packets for any reason, like placement at the network’s border.

To solve this problem, we propose to use Round Trip Time (RTT) as a measurement for the AED system. The monitoring module requests nodes for responses, and the AED’s detection engine can detect active eavesdroppers in WSNs based on the response delay. We focus on three potential monitoring systems for this measurement: static monitoring nodes, UAV-based monitoring, and neighborhood monitoring. To find the optimal places for static monitoring nodes, we utilize a Genetic Algorithm(GA), and to find the path of flight for UAVs for measuring RTT, we use Hamiltonian path planning. The simulation results indicate that the RTT-based AED system can detect active eavesdroppers regardless of their locations, with a high detection rate (≥ 90%) and a low false-positive rate (≤ 5%) and outstanding performance (AUC ≈ 0.97). In addition, we analyze and discuss the network overhead, advantages, and disadvantages of the in-band neighborhood monitoring system.
Acknowledgements

My sincerest appreciation goes to Dr. Irfan S. Al-Anbagi for providing me with the research opportunity and for his support and motivation. It has been an honor to work with him.

My sincere thanks to the members of my thesis committee: Dr. Nashid Shahriar and Dr. Paul Lafurge for their insightful comments.

My heartfelt gratitude goes to my beloved family for their unconditional support, care, and encouragement.
Dedication

To my beloved parents and to my dear brother.
# Contents

Abstract

Acknowledgements

Dedication

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 An Overview of Security Issues in WSNs

1.2 Passive Attacks in WSNs

1.3 Motivations

## 1.1 An Overview of Security Issues in WSNs

## 1.2 Passive Attacks in WSNs

## 1.3 Motivations
1.4 Objectives ....................................................... 9

1.5 Methodology .................................................... 9

1.6 Contributions .................................................. 10

1.6.1 Conference Paper ........................................... 11

1.6.2 Journal Paper ................................................. 11

1.7 Thesis Organization ............................................ 12

2 Related Work ..................................................... 13

2.1 Introduction .................................................... 13

2.2 Eavesdropping Mitigation Techniques ......................... 14

2.3 Eavesdropping Detection Techniques .......................... 19

2.4 Conclusion ....................................................... 22

3 Active Eavesdroppers Detection (AED) System in Multi-hop Wireless Sensor Networks using Intra-node Delay Measurement .................................................. 23

3.1 Introduction .................................................... 23

3.2 The AED system Architecture ................................ 24

3.3 Monitoring Module ............................................. 25

3.4 Intra-node delay for active eavesdropper detection ............ 28

3.5 Detection Engine Module ....................................... 29

3.6 Results and Analysis ........................................... 32
4 Round Trip Time (RTT) measurement for Active Eavesdropping Detection in WSNs

4.1 Introduction ................................................. 44

4.2 What is RTT? ................................................. 45

4.3 Why RTT is an important parameter for Active Eavesdropping Detection? ............................................. 46

4.4 The AED’s Architecture and Detection Engine Module using RTT measurement ............................................. 46

4.5 AED’s Monitoring Module to measure RTT ................. 47

4.6 Static Monitoring Nodes ........................................ 48

4.6.1 GA for Placement of Static Monitoring Nodes .......... 49

4.7 UAV-based Monitoring ........................................ 55

4.7.1 Path Planning for the UAV .................................. 57

4.8 Neighborhood Monitoring ...................................... 60

4.9 Results and Analysis ........................................... 60

4.10 Conclusion ...................................................... 69
5 Conclusions and Future Work

5.1 Conclusions .............................................. 71

5.2 Future Work ............................................. 72

References ................................................. 74
List of Figures

1.1 A sample taxonomy for WSNs applications [8] ............................ 4

2.1 A tree for eavesdropping defense in WSNs ................................. 15

3.1 The AED system Architecture...................................................... 24

3.2 A drone for monitoring intra-node delay parameter [12]............... 27

3.3 The Detection Module Flowchart.................................................. 30

3.4 ROC for the AED system for the variable window size for normal profile creation................................................................. 37

3.5 Change the location of an undetected active eavesdropper located in the network border............................................................ 39

3.6 ROC for the AED system, after relocating undetected active eavesdroppers................................................................. 41

3.7 AUC value for different size of time window of creation normal profile. 42

4.1 Measuring Round Trip Time in IP Network[59]. ............................ 45
4.2 Measuring Round Trip Time in IP Network with [59] 47
4.3 Flowchart for GA. 52
4.4 Chromosome representation. 53
4.5 Single point crossover. 54
4.6 An example for solving static monitoring placement problem in a WSN
with 200 nodes and 66 pre-defined locations for monitoring nodes in
an area of $100m \times 100m$ 56
4.7 UAV for RTT. 57
4.8 Hamiltonian path planning for the UAV in a grid WSN with 36 nodes. 58
4.9 DJI-FPV can fly with maximum speed of $39m/s$ [68]. 59
4.10 A grid WSN with 64 nodes, 16 active eavesdroppers in area of $200m \times
200m$ in the Cooja environment. 61
4.11 ROC for static monitoring nodes approach for measuring RTT. 62
4.12 ROC for the AED system with UAV-based monitoring with $20m/s$
speed. 64
4.13 ROC for the AED system with UAV-based monitoring with $30m/s$
speed. 65
4.14 ROC for AED system using neighborhood monitoring to measure RTT. 66
4.15 Average End-to-End delay. 67
4.16 Average Throughput. 68
4.17 Average Power Consumption.
List of Tables

3.1 Simulation Parameters. ............................................. 34
3.2 Confusion Matrix for evaluating the AED system. ............ 36
Chapter 1

Introduction

1.1 An Overview of Security Issues in WSNs

With the rapid advancement of microelectronics and wireless technologies, small electronic devices with minimal resources are now available at low cost, capable of gathering and processing information independently and flexibly. These devices are networked and scattered widely, forming wireless sensor networks (WSNs) [1]. WSNs are a sort of ad-hoc network that has been developed for a variety of purposes such as monitoring environments, tracking persons or equipment, and so on (please refer to Figure 1.1 that shows some of WSNs applications). WSNs, in particular, have grown fast since the introduction of sensors in the Internet of Things (IoT), and their utilization has surged tremendously [2].

WSNs consist of numerous tiny nodes that are randomly or planned spread across an environment in order to acquire specialized information. Temperature, humidity,
CO2 level, oxygen level, and other environmental data are some examples of sensing information measured by each node. Then they will be forwarded to the next node (hop), and with multi-hop communication, they will be delivered to the sink node or gateway, where the users can use them locally or over the Internet [3].

A wireless sensor node is a compact device that includes sensing and computing components, as well as a wireless transmitter and a small battery[4]. Therefore, the majority of WSN devices have specific limitations, such as an environment without a centralized administration and limited computation capacity and power, which result in threats that expose WSNs to potentially hazardous attacks. These limitations and the ad-hoc setting of WSNs cause implementing a wide range of standard security mechanisms on these devices to be not practical or too difficult. In recent years, threats such as compromises of the node, energy usage, and routing attacks have emerged. Using the node compromise, which is one of the most significant problems in WSN, an attacker can physically capture and compromise stored data or software. In addition, compromises in energy consumption and routing settings, which restrict the transfer of data without interruption, remain problems to be resolved [2].

Moreover, because of the broadcasting nature of wireless media, it is vulnerable to a wide range of passive and active attacks, particularly in the physical layer (PHY) and medium access control (MAC). A network attack is an attempt to compromise confidentiality, integrity, or availability. WSNs are well-known components of the
IoT network and are vulnerable to a variety of attacks. As a result, we traditionally categorize the attackers’ behavior into two major groups: Passive attacks and Active attacks. Passive attacks are deployed to get sensitive information from a network without any modification of data or packets [5]. However, active attacks typically change the data (modification, re-transmission, rejection of packets, etc.), compromise the communication between the nodes, and impair the nodes’ availability [6].

In other words, Passive attacks aim to grab sensitive information from a network without modifying it. Eavesdropping and traffic analysis are the best examples of passive attacks. In contrast, the active attacks category involves attackers altering the targeted network’s functions, operations, or packets. There are many active attacks in wireless networks, especially in WSNs and IoT, including black-hole, flooding, Denial-of-Service (DoS), jamming, Sybil, wormhole, sinkhole, etc. [7]. In this thesis, we focus on passive attacks because they are very hard to deal with, and there is not much research to address them.

1.2 Passive Attacks in WSNs

Passive attacks are challenging to detect due to the lack of radio signals emitted by the adversaries or any significant footprint. From the privacy perspective, passive attacks are obviously a serious concern. In passive attacks, the attackers frequently
Figure 1.1: A sample taxonomy for WSNs applications [8]

disguise themselves and break the communication link to collect data. Passive attacks include eavesdropping, node malfunction, node tampering/destruction, node interruption, and traffic analysis [4].

In general, Passive attacks can be divided into two main categories: Eavesdropping and Traffic Analysis. In eavesdropping attacks, eavesdroppers listen to transmitted packets within the networks to extract useful information from them. In Traffic Analysis, adversaries try to analyze network traffic parameters such as time, frequency, and intervals of packets in the networks to find some critical information like the
location of the sink [6].

1. **Eavesdropping**: Passive information gathering is a synonym for eavesdropping. It is possible to intercept classified information by tapping communication links. Wireless networks are therefore more vulnerable to passive attacks. Since WSNs use short-range communications, an attacker must be in close proximity to acquire useful information by eavesdropping; hence, WSNs are less susceptible to tapping than long-range wireless technologies. Interception of messages transmitted via WSNs may reveal the following useful information: the physical location of specific nodes, such as cluster heads, gateways, key distribution centers, etc.; message identifiers (IDs), timestamps, and other fields, as well as practically anything that is not encrypted [7].

2. **Traffic Analysis**: The traffic pattern of a network may be as valuable to adversaries as the content of data packets. By examining traffic patterns, one can obtain vital information about the network’s topology. In WSNs, the nodes closer to the sink, or base station, make more transmissions than the other nodes because they relay more packets than the nodes further from the sink. Similarly, clustering is an essential technique for scalability in WSNs, and cluster heads are busier than other network nodes. A denial-of-service (DoS) attack against these nodes or eavesdropping traffic headed for them may have a bigger impact if an adversary is able to detect the base station or cluster heads.
This type of useful information can be obtained by analyzing the traffic. In addition, traffic patterns can reveal additional sensitive data, such as activities and intentions. In tactical environments, silence may signal an imminent war, tactical maneuver, or penetration. Similarly, a rapid rise in traffic may signal the beginning of a planned attack or raid [7].

Passive attacks, according to the definition, do not cause changes in network packets or functionality. Passive attacks are challenging to detect because they have no effect on network functionality or parameters. However, detecting them is critical because attackers conduct passive attacks before launching targeted and damaging attacks. In the hacker world, this phase is known as Reconnaissance Phase [9].

1.3 Motivations

To reduce the severity of eavesdropping attacks, the first line of defense is to encrypt data using cryptographic techniques. Cryptographic algorithms, on the other hand, are based on some difficult mathematical problems, such as discrete logarithm problems. The developers of cryptographic algorithms believe that the adversary’s computation processing capability is restricted. However, this is not a reasonable assumption. Nonetheless, as quantum computers advance, many cryptographic techniques will be significantly threatened in the near future [10]. The discrete logarithm
problem, the elliptic-curve discrete logarithm problem, the integer factorization problem, and any other closely related mathematical problems are expected to be susceptible to being cracked by quantum computers [11].

Regarding the breaking of traditional cryptography by quantum computers, the author of the book [11], entitled "Cryptography Apocalypse: Preparing for the Day When Quantum Computing Breaks Today’s Crypto" and published in 2019, tried to answer this question: "When Will the Quantum Crypto Break Happen?" [11]. In brief, the most qualified computer scientists and the U.S. government agree that you should start preparing now, even though we do not yet know when quantum supremacy and the quantum crypto break will occur. For example, a consensus study report titled "Quantum Computing: Progress and Prospects" was released in 2018 by the U.S. National Academy of Sciences. According to the report, at least a decade must pass before quantum computers can break RSA-2048-bit encryption [11].

In cyberspace, the first essential step of every targeted attack is Reconnaissance Phase [9]. This step, which is commonly accomplished by a prelude of eavesdropping attacks, is critical and significantly impacts the success or failure of the following attacks. Furthermore, because wireless networks are broadcast, eavesdropping attacks are simple to carry out. Thus, it is highlighted the importance of the utilization of the second line of defense like eavesdropping detection systems [12].

Eavesdropping can be done by two different types of nodes in WSNs. The first
group is passive eavesdroppers that listen to the media stealthily without any permission and participation in the network process. The second group is active eavesdroppers, which are usually compromised nodes. They act like legitimate nodes and participate in network functionalities like forwarding packets. Most of the work in the literature has tried to mitigate the effectiveness of passive eavesdroppers. For instance, papers [7], [10], [13], [14] focused on how to decrease the probability of sniffing packets via eavesdroppers in wireless networks by using information theory, jamming, or power control techniques. However, these techniques, like power transmission control [13], may increase the probability of capturing packets by the active eavesdroppers, or even they are fruitless in the face of them [12].

It is difficult or impossible to detect passive eavesdroppers because they do not affect network parameters. However, active eavesdroppers are a member of legitimate nodes that relay packets and collaborate with other nodes in multi-hop WSNs. To the best of our knowledge, no work in the literature tries to use the second line of defense for active eavesdroppers in multi-hop WSNs. Therefore, we are motivated to cover this gap. We hypothesize that simultaneously performing eavesdropping and normal operation by active eavesdroppers impacts some parameters that we can take advantage of them to detect their malicious activity in WSNs. Thus, this thesis focuses on detecting active eavesdroppers in a homogeneous multi-hop WSN. Active Eavesdroppers Detection (AED) system is proposed for detecting the presence of
active sniffers in WSNs or IoT. The system includes a monitoring module that gathers measurements from nodes and sends them to the detection engine module.

1.4 Objectives

This thesis’s main objectives can be summarized as follows:

- Develop a system to detect active eavesdroppers for homogeneous multi-hop WSNs in tactical environments.

- Design a modular architecture for the detection system considering the constraints of WSNs.

- Study and examine related features or measurements that reveal the presence of active eavesdroppers in multi-hop WSNs.

1.5 Methodology

Since the main objective of this thesis is to develop an intrusion detection system to detect active eavesdroppers in WSNs, we follow the Anomaly Detection methodology. So, we follow the following steps:

1. Selecting specific feature(s) or measurement(s) related to the attack or the attacker.
2. Developing a system to measure or monitor the feature(s) or measurement(s) or assuming how to collect or measure them.

3. Choosing an appropriate anomaly detection algorithm that detects any deviation from a normal profile of the feature(s), including statistical-based, Machine Learning-based, Artificial Intelligence-based, etc.

4. Testing and evaluating the system based on a dataset (including the selected features) regarding acceptable performance metrics in the literature. If there is no available dataset, we need to create it using simulation, emulation, or gathering from the real world.

1.6 Contributions

The main contributions of this thesis are summarized below:

- We propose a novel system that can detect active eavesdroppers in multi-hop homogeneous WSNs with a high detection rate and low false positive rate.

- The AED system utilizes a monitoring system that uses a non-intrusive or out-of-band method to measure the intra-node delays or RTT. It is not necessary to synchronize monitoring nodes or UAVs.

- The AED system has a modular architecture that can be expanded easily in the future.
• The lightweight algorithm (using Z-score) for the detection engine allows us to run it on edge devices.

• We utilize a Genetic Algorithm (GA) for static monitoring nodes placement.

• To find the path of flight for UAVs for measuring RTT, we use Hamiltonian path planning.

• We compare three different monitoring systems for measuring RTT, including the non-intrusive or out-of-band method and also the in-band method.

1.6.1 Conference Paper


1.6.2 Journal Paper

1.7 Thesis Organization

The structure of this thesis is as follows. In Chapter 2, we briefly review related works of eavesdropping attacks in the literature. Chapter 3 is devoted to the design of the AED system, its architecture, and modules using intra-node delays of nodes. Chapter 4 discusses another measurement for detecting active eavesdroppers in WSNs and different monitoring approaches for measuring RTT. Finally, the conclusion and future works are presented in the last chapter of this thesis.
Chapter 2

Related Work

2.1 Introduction

In this chapter, we review the work related to defense techniques against eavesdropping in networks, namely Mitigation techniques and Detection techniques. Mitigation methods are usually utilized to reduce the impacts of eavesdropping attacks. These methods are known as the first layer of defense. Moreover, detection techniques like Intrusion Detection Systems (IDS) are the second line of defense. In order to review the defense techniques against eavesdropping, we develop a tree (see Figure 2.1). We review related papers in the literature for these techniques in the following subsections.
2.2 Eavesdropping Mitigation Techniques

Cryptographic algorithms are the preliminary technique for protecting networks against eavesdropping, but they have two significant drawbacks for WSNs and IoT. Firstly, they usually make high computational costs and communications overhead. Secondly, key distribution in ad-hoc and constraint networks is a challenging problem. Recent researches on designing cryptosystems compatible with constraints of WSNs or IoT exist in the literature, such as [15]–[18]. However, these days conventional cryptosystems are threatened by the advancement of Quantum Computers [19]. Although scholars are working on designing post-quantum cryptosystems, which resist quantum computers, there are many associated problems, including standardization, large key sizes, slow key generation and, etc., that should be tackled first to deploy in WSNs or IoT [19].

Information-theoretic techniques make a hindrance for eavesdroppers to create wiretaps in communications channels. The fundamental concept of this physical layer technique was firstly introduced by Claude Shannon [20]. Although Shannon focused on symmetric-key cryptosystems, Aaron Wyner worked on the wiretap channel, which taught us that secrecy could be provided by the communications channel itself without any encryption [21]. Therefore, we can exploit some inherent features of the physical layer in wireless channels like diffusion, superposition, fading, interference, and path diversity (exploiting in Multiple Input Multiple Output system) to achieve
Figure 2.1: A tree for eavesdropping defense in WSNs
In other words, we want basically to make it difficult for potential eavesdroppers to gain access to confidential messages. For instance, He and Yener [22] proposed an information-theoretic method to secure multi-hop communications with eavesdropping relay nodes using nested lattice code. However, this method imposes computational processing on each node, making it unsuitable for resource constraint networks.

Friendly jamming or making artificial noise is another technique to mitigate eavesdropping attacks in wireless networks. The main concept is that friendly jammers can assist legitimate nodes in sending their messages confidentially. Meanwhile, the artificial noise disrupts the receiving channel of malicious eavesdroppers. Wang et al. [23] introduced a UAV-enabled friendly jamming system to preserve the confidentiality of industrial IoT by emitting artificial noise. It reduces the probability of eavesdropping attacks. However, the major drawback of friendly jammers might be reducing signal to noise plus interference (SINR) for legitimate users, leading to retransmission and high power consumption.

In the beamforming technique, legitimate nodes send the signal directly to the destination via special smart antennas. This technique tries to minimize the chance of eavesdroppers receiving confidential information [20]. Zhang et al. [24] investigated an optimization algorithm to reduce the exposure region to combat passive eavesdroppers in wireless networks. However, the optimal beamforming algorithm relies on knowing

16
the receiver’s location by the sender. In addition, deploying smart antennas as well as running optimization algorithms on tiny and cheap wireless sensor nodes may be infeasible.

In the transmission power control method introduced by Kao and Marculescu [13], the basic idea is that nodes can tune the transmission power or transmission range in order to minimize the probability of eavesdropping risk in MANETs. This method also has some advantages extra to increasing the level of security. It can improve throughput, energy conservation, QoS support, spectrum reuse, and reduce interference [13]. The core idea is that if the source node uses high transmission power to send packets to the destination, passive eavesdroppers can easily hear and capture packets. However, if it uses low-level power to send, the packets will relay by adjacent nodes, but the eavesdroppers are not able to overhear the transmission. In this method, the problem is that they assumed the eavesdroppers are passive and they do not relay any packets. Moreover, we can also mention other drawbacks of this method. It can increase the probability of low SNIR in the receiver that causes retransmission of packets. Moreover, depending on the topology of the network and the location of eavesdroppers, it leads to increasing the hop count in ad-hoc networks, so it may increase the chance of packet capturing by active or passive eavesdroppers. It is a kind of dilemma.

Secure routing, as well as anonymous routing protocols, can be used to defend
against eavesdropping, but these also mostly depend on cryptographic algorithms. Secure routing protocols, or in a broad view, anonymous routing protocols, can protect wireless networks against compromising network operations by adversaries. For instance, adversaries can maliciously change a route to access the information by some compromised nodes or sniffers. Therefore, secure routing protocols are responsible for protecting networks from such Byzantine behavior [13], [25]. There are many works in the literature for secure routing protocols in Mobile ad-Hoc Networks (MANETs), WSNs, and IoT like [26]–[30]. However, we can also enumerate the disadvantages of cryptographic-based mitigation methods as producing high overhead for networks, increasing processing delays in nodes, increasing the size of packets in some cases, and as a result, increasing power consumption for nodes.

Game-theoretic approaches have recently gained the attention of some researchers to solve security issues. By modeling interactions between attackers and defenders with game theory, researchers have tried to gain advantages of mixing techniques like friendly jamming and power transmission control. Wang et al. [31] presented an anti-eavesdropping game for selecting friendly jammers between relaying nodes in wireless networks using the Bertrand game. Their approach consisted of forming a Bertrand game based on price competition that allocates power among the selected nodes to achieve maximum secrecy capacity.
2.3 Eavesdropping Detection Techniques

The second layer of defense is deployment detection techniques in the networks. From a broad perspective, there are two major methods to detect sniffers or eavesdroppers in all types of networks [14]:

**Challenge-based**

In this method, the detection is based on the response of suspicious nodes in networks. In order to do this, we can provoke or stimulate the suspicious nodes by sending them some special packets. For example, in a normal condition, nodes usually do not respond to packets that are not destined to them, but when they are in sniffing mode, they might respond to unrelated packets wrongly [14].

In other words, the challenge-based Detection technique uses the fact that there is a probability that sniffers are not completely passive, so they may respond to some particular types of packets. Sniffers or eavesdroppers need to use a special mode in their Network Interface Controller (NIC). Monitor or promiscuous mode has to be activated in order to listen to the media and capture transmitting packets within networks. In this method, detectors usually use packets with forged or fake MAC addresses. The attackers captured the packets, but normal nodes will drop them immediately. Since there are some flaws or bugs in the kernel of sniffers, they will send the captured packets into the above layers. Then the OS responds to them.
wrongly, so the detectors are monitoring their responses, and if there is a response for the challenged packets, the detectors will send an alarm of sniffing detection [14], [32].

Examples of particular types of packets for the challenge-based method:

- **ARP**: This protocol is used to find addresses of known MAC addresses. It broadcasts packets to find the targeted IP. In sniffers detection systems, detectors create ARP request packets with fake MAC addresses and valid IP addresses. As sniffers do not apply any filters on received packets, the sniffers, after receiving the packets, will send an ARP reply. It can reveal that the suspicious node is an eavesdropper [32].

- **ICMP**: It is utilized like ARP packets. The detector sends an ICMP packet with a correct destination IP address but with forged destination MAC addresses. The sniffers reply to this packet wrongly, and we can detect the sniffer [32].

- **DNS**: Using DNS packets is another way to detect sniffers. Since sniffers usually perform a reverse DNS lookup to find the name of unknown IP addresses, detectors can take advantage of this protocol. In this technique, the detector sends some packets with unused IP addresses. If the suspicious nodes generate and send reverse DNS packets in the networks to find the name of the unknown addresses, the detector can say that the sender is a sniffer [14], [32].
• FTP: The detector sends some login session information, in plaintext format, for an FTP service in the networks. If the fake credential is used for login into the FTP server, the detector can say that there is a sniffer in the network [14], [32].

Measurement-based

Eavesdroppers or Sniffers in networks listen to the media and receive a huge amount of packets. As a result, the traffic load of the attackers may be so high. Therefore, measuring some related parameters would be helpful in screening the malicious nodes. For instance, Round-Trip Time (RTT) can be a useful measurement in wired networks [14].

The Authors in [14] discussed that using the aforementioned challenges for the challenge-based method is useless currently because the sniffers have enhanced both hardware and software to be totally passive, and they do not respond to the mentioned challenges anymore. Therefore, they proposed a measurement-based system to detect sniffers in wired networks. The system utilized a probing host to send (using macof tool) many packets to the suspected host and then measure two related parameters. The first parameter is RTT (using ping) and the second parameter is the rate of downloading a file (using curl tool). Then, the parameters are fed to ML algorithms (XGBoost and LightGBM) to detect the sniffers.
Shen and Huang [33] claimed that they proposed the first eavesdropping detection system for wireless networks called EarFisher. The proposed system uses a stimulating technique along with a detector that employs signal processing algorithms to sense and analyze Electromagnetic Radiation (EMR) of the memory of eavesdroppers. They used the fact that eavesdroppers write packets in their memory even though they are not targeting them. They deployed the system in a testbed for a WiFi network. The system detects eavesdroppers based on an extra process that eavesdroppers do internally (writing the packets in their memory that have nothing to do with them). However, wireless sensor nodes have weak hardware with limited resources, so the EMRs of the nodes are very weak to be detected by EarFisher [12].

2.4 Conclusion

To the best of our knowledge, there is no active eavesdroppers detection system for WSNs. The few proposed systems in the literature for wireless networks are not compatible with the special characteristics of WSNs. To cover this gap, in this thesis, we propose a novel system for detecting active eavesdroppers in homogeneous multi-hop WSNs using a measurement approach.
Chapter 3

Active Eavesdroppers Detection (AED) System in Multi-hop Wireless Sensor Networks using Intra-node Delay Measurement

3.1 Introduction

In this chapter, we present our proposed AED system. Scattering the wireless nodes in tactical environments increases the risk of node vulnerability because nodes are exposed to physical access by adversaries. With the AED system, we can detect the presence of active eavesdroppers in WSNs. The AED system [12] and its components, including the architecture, the monitoring module, and the detection engine,
3.2 The AED system Architecture

The AED system consists of two main modules. 1) A monitoring module measures intra-node delays with an out-of-band or passive monitoring tool. The measurements of the monitoring module are fed into a detection engine module. 2) A detection engine module utilizes a statistical-based anomaly detection method applied to the measurements. The output is a list of suspicious nodes considered potential active eavesdroppers in the networks [12].
The architecture for the AED system is shown in Figure 3.1. We assume that a homogeneous multi-hop WSN is deployed in an open tactical environment to collect sensitive data. The sensor nodes send information via multi-hop communications to the WSN gateway (WSN-GW). A monitoring network measures and gathers related parameters for the AED system modules and sends them to the monitoring gateway (Monitoring-GW). The monitoring module can benefit from both static monitoring nodes (gray circles in Figure 3.1) or different types of UAVs. The detection engine of the AED system runs on the WSN-GW [12].

### 3.3 Monitoring Module

The monitoring module is responsible for measuring related parameters from the deployed WSN. The monitoring module in the AED system plays a pivotal role because the detection is made based on the provided measurements. It provides intra-node delay measurement for the detection engine module. In addition, this module should be robust in tactical environments where adversaries can compromise the nodes relatively simply. Therefore, selecting an efficient and secure method is essential in terms of accuracy and effectiveness. In ad-hoc networks, there are two major monitoring architectures, including active or in-band and passive or out-of-band architecture [34], [35].

- **Active or in-band architecture**: In this architecture, target nodes measure
or gather monitoring data and send them to the sink using extended packets’ fields or special packets. Network management protocols for WSNs, like Sensor Network Management Protocol (sNMP) and Sensor Network Management System (SNMS), were designed for this purpose[36]. This architecture is not appropriate for security monitoring mechanisms due to the high risk of modification and falsification by adversaries in tactical environments. Another disadvantage of this architecture is that it is not applicable in currently deployed networks due to the necessity of changing running codes or software on the wireless nodes [35], [37].

- **Passive or out-of-band architecture**: On the other hand, in passive or out-of-band architecture, passive sniffers gather target information from wireless nodes and send them to the sink or gateway. It is a more appropriate architecture to measure metrics regarding security monitoring [34]. Moreover, it can be implemented in deployed WSNs without changes in OS or software of deployed networks. The monitoring module can use drone-assisted sniffers to monitor the networks in open environments [38], [39], especially for limited accessible areas in tactical environments (please see Figure 3.1).

In the AED system, intra-node delay measurement can be captured by the passive architecture effectively because active sniffers cooperate in packets forwarding. Figure 3.2 shows a UAV equipped with two wireless interfaces used in our system.
The first one is for monitoring the channel of the target WSN, and the second one is for out-of-band communications between monitoring wireless nodes or UAVs (See both Figure 3.1 and Figure 3.2). In addition, static monitoring nodes equipped with two wireless interfaces can also be used to monitor the target WSN in the same way as UAVs. This type of monitoring node is shown in Figure 3.1 (the gray nodes). In addition, in terms of designing dual interfaces for wireless static monitoring nodes, the paper [40] discussed how to design a switch agent for radio compatible with both IEEE 802.15.4 and IEEE 802.11 standards.

Figure 3.2: A drone for monitoring intra-node delay parameter [12].

To measure intra-node delay, for instance, in Figure 3.2, there are three nodes
where the packet \( P1 \) originated from node \( i \), relayed by node \( j \), and destined to node \( k \). If the monitoring node or UAV listens to the media, it can measure the intra-node delay in \( j \) by capturing and timestamping the \( P1 \). Synchronization is not necessary for this monitoring module. The measurements are sent via the out-of-band interface to the monitoring gateway. In collaboration with the main gateway, it provides measurements to the detection module. In terms of the coverage problem of static monitoring nodes or UAVs, existing algorithms or methods in the literature can be used for scattering static nodes or hovering UAVs [37], [41]. This area is beyond the scope of this thesis. Nevertheless, we assume sniffer placement algorithm [37] is used for scattering static monitoring nodes and also for hovering (static) UAVs [41].

3.4 Intra-node delay for active eavesdropper detection

We assume that when active eavesdroppers run processes like sniffing or eavesdropping, their intra-node delays become higher than normal relaying nodes. The authors in [42] presented experimental results for intra-node delays in ad-hoc networks. The results showed that when relaying nodes carry out more processes in ad-hoc networks, their intra-node processing delays increase significantly. Although intra-node delays are negligible in traditional wired networks, it is higher in WSNs due to the constraint resources of the sensor devices. In articles [42]–[45], the authors
discussed the problem of intra-node delays in WSNs and highlighted that the intra-node delays are not negligible for these networks. Especially in nodes with activated promiscuous mode (i.e., turning off the hardware filtering in the MAC layer), the time that packets stay in the queue of the MAC layer will increase as they capture nonrelated packets in the media. Moreover, capturing and copying unintended traffic leads to CPU interruptions leading to a higher processing delay. Therefore, the AED system uses this feature to detect active eavesdroppers [12].

3.5 Detection Engine Module

The detection engine module of the AED system takes advantage of a statistical-based anomaly detection algorithm. Using the statistical information provided by the monitoring module, a Z-test-based statistical analysis is used to detect the presence of active eavesdroppers in the network. The Z-test determines whether or not the hypothesis regarding the network measurements is valid by comparing two groups of measurements [32], [46].

Firstly, to detect active eavesdroppers, the detection engine of the AED system needs to create a normal profile from intra-node delay measurements of legitimate nodes, which are provided by the monitoring module. The normal profile can be created by collecting intra-node delay measurements from nodes that are not malicious. The AED system uses the detection engine module described in both Figure 3.3 (as
null hypothesis \( (H_0) \): The average of the intra-node delays of normal nodes and active eavesdroppers are equal.

- **Alternative Hypothesis \( (H_1) \):** The average of the intra-node delays of normal nodes and active eavesdroppers are not equal.

In Algorithm 1, \( \bar{X} \) is the average of the measured samples, \( \sigma \) is the standard
Algorithm 1 Detection Engine Module

Inputs: A set of measurement samples ($n_{np}$) from nodes to create a normal profile,

A set of measurement samples for each node ($n_{ID,t}$) that is collected from the network for testing,

$Z$: z-score threshold for making a decision.

Output: A list of suspicious nodes

Initialisation: CALC ($X_{np}, \sigma^2_{np}$) of the normal profile

1: for each node that has measurement samples do
2: CALC ($X_{ID,t}, \sigma^2_{ID,t}$) for each node
3: CALC z-score: $z_{ID} = \frac{(X_{ID,t} - X_{np})}{\sqrt{\frac{\sigma^2_{np}}{n_{np}} + \frac{\sigma^2_{ID,t}}{n_{ID,t}}}}$
4: if $z_{ID} > Z$ then
5: ADD the node ID to the suspicious list
6: end if
7: end for
8: return the suspicious list of the node ID

deviation, and $n$ is the number of measurements. We assume that there are $N$ nodes in the network, and each node has an $ID \in \{1, 2, 3, ..., N\}$. The detection engine module firstly forms a normal profile ($X_{np}, \sigma_{np}$) from $n_{np}$ intra-node delay measurement samples, and then for each node calculates ($X_{ID,t}, \sigma_{ID,t}$) from $n_{ID,t}$ measured samples. After that, it computes z-score ($z_{ID}$) using formula 3.1. Based on the z-score and the defined threshold ($Z$), the detection module decides whether node $ID$ should be added to the suspicious list or not. It is worth noting that edge devices that run this detection engine can save just the pair of $X_{np}$ and $\sigma_{np}$ of the normal profile. It can reduce the memory usage of edge devices. Also, in terms of computation process, calculating $X_{ID,t}$ and $\sigma_{ID,t}$ in a predefined time window is a
lightweight process.

\[ z_{ID} = \frac{(X_{ID,t} - X_{np})}{\sqrt{\sigma^2_{2p} + \sigma^2_{1D,t} \frac{1}{n_{ID,t}}}} \]  

(3.1)

3.6 Results and Analysis

In this section, we present the evaluation process of the AED system. To simulate a multi-hop homogeneous WSN and add active eavesdroppers to the network, we use Contiki OS and its Cooja simulator [47] for implementation and simulation.

Researchers usually use publicly available datasets for evaluating their proposed intrusion detection systems for wired networks [48]. However, unfortunately, the available datasets for WSNs, like WSN-DS [49], do not contain active eavesdropping attacks and the intra-node delays feature. Therefore, utilizing simulator software to construct the dataset is a fast and cost-effective solution. First, we simulate a WSN in the Cooja environment with simulation parameters in Table 3.1.

The Cooja simulator can generate a Mote Output log file containing the event logs of network nodes. As a result, in order to record intra-node delay measurements, we modify motes’ codes to make an event in the motes’ output when a node receives a packet that must be forwarded to neighbor nodes. In addition, when the relaying nodes forward the received packets, another event must be generated by the forwarder.
The simulation’s Mote Output log file is fed into Python scripts, which calculate intra-node delays and prepare the dataset. The outputs are similar to the measurements of the monitoring module for the intra-node delays. To construct the dataset, we run the simulation 20 times with different random seeds. Moreover, we implement the detection engine module in MATLAB environment to evaluate the AED system [12].

Regarding the intra-node delay measurements, as we expect and hypothesize in section 3.4, we observe that in the simulation results with parameters in Table 3.1, the average of intra-node delay measurements for active eavesdroppers is about 12% higher than normal nodes. In this case, they are about 50\(ms\) for normal nodes and 56\(ms\) for active eavesdroppers on average. It is clear that the measurements are depended on the size of the network, mote hardware, type of protocols in different layers, the load of traffic, etc.

In terms of capacity and specification of wireless motes, the Zolertia Z1 mote has a second-generation MSP430F2617 low-power microcontroller that has an effective 16-bit RISC CPU running at 16MHz, built-in clock factory calibration, 8KB RAM, and 92KB Flash memory. It is also equipped with the well-known CC2420 transceiver, which runs at 2.4GHz and has an effective data rate of 250Kbps. It complies with IEEE 802.15.4 [50].
Table 3.1: Simulation Parameters.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>100m * 100m</td>
</tr>
<tr>
<td>Packet Size</td>
<td>80 Bytes</td>
</tr>
<tr>
<td>Packet Generation Interval</td>
<td>60 sec.</td>
</tr>
<tr>
<td>Physical Layer</td>
<td>IEEE 802.15.4</td>
</tr>
<tr>
<td>MAC Layer</td>
<td>CSMA (Non-beacon enabled IEEE 802.15.4)</td>
</tr>
<tr>
<td>Network Layer</td>
<td>RPL (IETF RFC 6550-2012)</td>
</tr>
<tr>
<td>Transport Layer</td>
<td>UDP</td>
</tr>
<tr>
<td>Adaptation</td>
<td>6LoWPAN</td>
</tr>
<tr>
<td>Hardware Platform</td>
<td>Zolertia Z1 mote</td>
</tr>
<tr>
<td>Number of Nodes</td>
<td>50 normal nodes + 10 active eavesdroppers</td>
</tr>
</tbody>
</table>

3.7 Simulation/Emulation Environment

A wide verity of simulators are available for WSNs and IoT, including OMNET++ [51], NS-2 [52], NS-3 [53], QualNet [54] Cooja [47], and so on. We select the Cooja simulator to generate a dataset for evaluating our proposed system. Cooja is a java-based simulator/emulator that is part of the Contiki OS. As opposed to traditional simulators like NS-2 [52], or NS-3 [53], Cooja is based on the full Contiki source code, and it emulates the device hardware to achieve near-realistic results,
especially for intra-node delay [43]. It enables the rapid deployment of the simulated experiments directly over the real motes’ hardware.

### 3.8 Evaluation Metrics

To evaluate the performance of the AED system, we form a Confusion Matrix (Table 3.2), and then draw Receiver Operating Characteristic (ROC), and calculate Area Under the ROC Curve (AUC). False Positive Rate (FPR) and True Positive Rate (TPR), Detection Rate (DR), or Sensitivity are two primary performance metrics (see equations 3.2 and 3.3) that are used to develop ROC and compute the corresponding AUC [55]. The AUC is a metric that indicates how well a binary classifier or IDS can predict labels. When AUC is equal to one, the system is perfect. Any AUC value less than 0.5 means that the system performance is poor. If AUC is above 0.9 indicates that the system performance is excellent [56].

\[
FPR = \frac{FP}{TN + FP} \quad (3.2)
\]

\[
TPR = DR = \frac{TP}{TP + FN} \quad (3.3)
\]
Table 3.2: Confusion Matrix for evaluating the AED system.

<table>
<thead>
<tr>
<th>Predicted</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Eavesdroppers</td>
<td>True Positive (TP)</td>
</tr>
<tr>
<td>Normal Nodes</td>
<td>False Positive (FP)</td>
</tr>
<tr>
<td></td>
<td>False Negative (FN)</td>
</tr>
<tr>
<td></td>
<td>True Negative (TN)</td>
</tr>
</tbody>
</table>

3.9 Deployment Scenario and Results

We test two scenarios to evaluate the effectiveness of the AED system. In the following subsections, we describe the scenarios, show the results, and analyze them.

Scenario 1

In this scenario, we scatter randomly 50 normal nodes and ten active eavesdroppers in an area of 100×100 meters, surrounding a sink node. Also, ten active eavesdroppers are randomly placed in that environment. The rest of the simulation parameters are set according to Table 3.1. We also change the size of the time window of creating the normal profile for the detection engine module to get the results.

As Fig 3.4 shows, we can achieve $DR = 70\%$ and $FPR < 5\%$ with AEDS. Increasing the length of the time window that is used to collect intra-node delay
measurements shows some changes in ROCs with a trend of increasing AUC values. Moreover, for instance, we can see that the ROC for the time window equal to 60 minutes has $DR = 70\%$ and $FPR < 5\%$. Also, we can tune the z-score threshold in Algorithm 1 to meet the level of security requirements. For instance, in highly sensitive environments that require a high level of confidentiality, we can decrease the z-score threshold so that the system would be labeled more nodes, including malicious
and benign, as active eavesdroppers (i.e. increasing FPR).

**Scenario 2**

As mentioned in the first scenario, we can achieve a DR of $DR = 70\%$. However, to increase the detection rate, we analyze the network topology and notice that active eavesdroppers located at the border of the network cannot be detected. The reason is that they do not relay packets, so the monitoring module cannot measure their intra-node delays. It is noteworthy to mention that the active eavesdroppers in scenario 1 are randomly scattered in the network. As a result, some of them are located at the border of the network. However, from an adversaries point of view in the real world, eavesdroppers have the intelligence to attempt to put themselves in places to capture transmitting packets as much as possible (i.e., between relaying nodes or close to the sink; we can assume that an insider reveals the location of the sink). Therefore, changing their locations is not an unrealistic scenario. In this scenario, we manually change the location of the undetected active eavesdroppers in a way to relay packets of the neighbor’s nodes and then rerun the simulation to evaluate our system. It is clear that the relocation of the undetected active eavesdroppers is not a solution to deal with the problem. However, in this scenario, we assume that the topology of the networks allows the monitoring module to measure the intra-node delays of the attackers.
Figure 3.5: Change the location of an undetected active eavesdropper located in the network border.

Figure 3.5 shows an example of the relocation of an undetected active eavesdropper in a section of the network. Figure 3.5 shows three normal nodes ($N_{12}$, $N_{22}$, and $N_{33}$), a sink, and an active eavesdropper ($E_1$). The AED system fails to detect $E_1$ because of its location and, as a result, inaccessibility to its intra-node delay. After such relocation, we extract the results according to Figure 3.6. As you can see, in comparison to scenario 1, the detection rate increases from 70% to 100% for the window time of 60 minutes because the monitoring module of the AED system can measure the intra-node delays of the relocated active eavesdroppers. Furthermore, such ROC curves help network administrators to tune the DR and FPR of the AED system [12].
In Figure 3.7, we test the efficiency of the AED system by varying the duration of the time window used to create the normal profile. The values of AUCs gradually increase to reach about 0.96 for the time window of 60 minutes. After that, we see a slight decrease in AUC values for the time window higher than 60 minutes. It shows that providing many intra-node delay measurements is not beneficial for the AED system because of the over-fitting problem. Therefore, fine-tuning the length of the time window for normal profile creation is crucial for such systems [12].

For clarification, the ROC graphs show stepped line manners because we have 50 normal nodes and 10 active eavesdroppers in these scenarios. Therefore, in TPR formula (refer to equation 3.3), the denominator is equal to 10, so TPR becomes a factor of 0.1 for the variable thresholds of Algorithm 1. As a result, TPR values belong to this set: \{0, 0.1, 0.2, 0.3, 0.4, ..., 0.9, 1.0\}. It is normal in ROC graphs [57].
3.10 Conclusion

In this chapter, we proposed a novel system for detecting the presence of active eavesdroppers in homogeneous multi-hop WSNs. The proposed system consists of two essential components: monitoring and detection engine modules. The monitoring module uses an out-of-band approach to measure intra-node delays of nodes as
a key feature for detecting active eavesdroppers in WSNs. This module can utilize asynchronous static monitoring nodes or UAVs for areas with low accessibility in tactical environments. In addition, the proposed architecture can be deployed easily for already established WSNs without any modification, and it is robust to the falsification of malicious nodes. The Z-test anomaly detection algorithm is used in the detection engine module. It is a lightweight algorithm for running on edge devices. Extensive simulations showed a high detection rate (100%) and a low false-positive
rate ($\leq 5\%$) with excellent performance ($AUC = 0.96$) after the relocation of nodes located at the border of the network. In the next chapter, we discuss another feature or measurement that helps the AED system for detecting active eavesdroppers regardless of their positions.
Chapter 4

Round Trip Time (RTT) measurement for
Active Eavesdropping Detection in WSNs

4.1 Introduction

In the previous chapter, we saw that using intra-node delay measurement as a feature to detect active eavesdroppers in WSNs has some drawbacks, especially for nodes located at the border of networks or nodes which do not relay packets because of selfishness or other reasons. This chapter discusses how to use RTT as a measurement to detect active eavesdroppers in WSNs. Moreover, we discuss different monitoring system architectures to measure RTT in tactical environments.
4.2 What is RTT?

RTT or round-trip delay time (RTD) shows how long it takes a network request to travel from a starting point of the network to the target point and then back to the starting point. It is usually measured in milliseconds (ms) [58]. In Figure 4.1, if we assume three wireless sensor nodes, the time that takes a request packet (i.e., ICMPv6.ECHO_REQUEST) to be sent to the target node and then receive the reply (i.e., ICMPv6.ECHO_REPLY) by the source node called RTT. Since the reference point is located at the source node, measuring RTT does not need synchronization [59], [60].

![Figure 4.1: Measuring Round Trip Time in IP Network][59].
4.3 Why RTT is an important parameter for Active Eavesdropping Detection?

The authors in [14] discussed the relationship between the RTT of control packets like Internet Control Message Protocol (ICMP) and the response time of hosts in promiscuous mode in wired networks. They designed a sniffer detection system by measuring the response time of nodes via ICMP packets. They noted that the reason for using this feature in the detection system is that sniffers’ network cards have to be activated in promiscuous mode. Thus, this mode causes a higher load of network traffic for the Network Interface Cards (NICs) and, as a result, it increases both the response time and intra-node delay parameters due to processing a higher number of packets in their hardware and software.

4.4 The AED’s Architecture and Detection Engine Module using RTT measurement

In sections 3.2 and 3.5, we proposed an architecture and detection engine module to detect active eavesdroppers in WSNs. We can use other related measurements to detect active eavesdroppers instead of using intra-node delay measurement. Thus, we use the same detection engine module as section 3.5, but the input measurement is RTT. Regarding this change, we need to redesign the monitoring module of the AED
system. In the following sections, we review three different approaches to measuring RTT.

4.5 AED’s Monitoring Module to measure RTT

In order to measure RTT, we need to measure this feature of scattered nodes in the tactical environments. While the RTT can be measured by sending \textit{ICMPv6.ECHO REQUEST} from WSN-GW to the target node and measuring it, but it is not reliable because some active eavesdroppers may be located as relay nodes and impact the measurement. For example, if we assume three wireless sensor nodes in Figure 4.2, the intermediate node is an active eavesdropper, which causes an extra delay in relaying packets, so the RTT measurement is not accurate for using in AED system.

![Diagram of measuring round trip time](image)

Figure 4.2: Measuring Round Trip Time in IP Network with [59]
As a result, we need to make the measurement in a way that the source node is the only one-hop distance from the target node. Therefore, we can select three different approaches to achieve the goal as follows:

1. Static Monitoring Nodes

2. UAV-based Monitoring

3. Neighborhood Monitoring

In the following subsections, we discuss each approach in more detail.

4.6 Static Monitoring Nodes

Regarding the monitoring module of the AED system to measure RTT, we need to determine the location of static monitoring nodes such that each target (monitored) node has a one-hop distance from the monitoring node and also be monitored by at least one static monitoring node. The main objective is to minimize the number of static monitoring nodes. So we can formulate the problem as an Integer Linear Programming (ILP). If there are some predefined or candidate positions in the WSN environment for the static monitoring nodes, so we can model it and solve it by some meta-heuristic approach like the Genetic Algorithm (GA) [61], [62]. Since this problem is an NP-complete, using analytical solutions or some commercial optimization tools like CPLEX [63] or Gurobi [64] is not reasonable for large-scale networks to
get optimal solution \cite{61, 65}. In the following subsection, we explain how to use the GA to find the optimal location of static monitoring nodes to measure RTT. We assume that the static monitoring nodes are equipped with two radio interfaces (one for measuring RTT, another for reporting the measurements to the Monitoring-GW) like the architect in section 3.2.

4.6.1 GA for Placement of Static Monitoring Nodes

Firstly we model the problem of placement of static monitoring nodes in WSNs to measure RTT.

System Model:

We assume that we have a homogeneous WSN. The problem is that given $x$ number of pre-defined positions for static monitoring nodes and $n$ number of wireless sensor nodes, we have to place a minimum number of static monitoring nodes to measure RTT of on-hop distance neighbors. There are some pre-defined potential positions for static monitoring nodes where we can place static monitoring nodes to provide measurements of RTT to the Monitoring-GW. Noted that in order to find the potential or candidate positions of monitoring nodes, we can use some algorithms like “Determine Candidate Monitor Locations” in article \cite{65}. However, for simplification, we assume that there are some pre-defined locations for potential positions of static
monitoring nodes. It is also assumed that the static monitoring nodes have out-of-band radio communication to transfer measurements to the monitoring-GW. So, the formulation is based on the paper [61]:

1. The set of target nodes (wireless sensor nodes) is $S = \{s_1, s_2, \ldots s_n\}$.

2. The set of static monitoring nodes is $M = \{m_1, m_2, \ldots m_k\}$.

3. The set of pre-defined potential locations of monitoring nodes is $L = \{l_1, l_2, \ldots l_x\}$.

4. $R$ depicts the sensor nodes’ communication range.

5. $d(s_i, m_j)$ shows the Euclidean distance between $s_i$ and $m_j$.

6. The potential spots that fall inside $s_i$’s range are shown in $C(s_i)$.

$$C(s_i) = \{l_j | d(s_i, l_j) \leq R \text{ and } l_j \in L\}. \quad (4.4)$$

Therefore, we need to place a monitoring node $m_r$ on one of the pre-defined potential locations from $C(s_i)$ to ensure monitoring of $s_i$ [61].

7. $W$ denotes the set of monitoring nodes, which can monitor the whole target wireless sensor nodes and $\omega_i$ denotes the set of monitoring nodes that can monitor $s_i$. In other words, $\omega_i = \{m_r | d(s_i, m_r) \leq R, \forall r, 1 \leq r \leq k\}$ and so

$$W = \bigcup_{i=1}^{n} \omega_i \quad (4.5)$$
Linear Programming (LP) form for the static monitoring problem:

The objective of the static monitoring problem is to minimize the number of monitoring nodes. Let $l_i$ be a Boolean variable:

$$l_i = \begin{cases} 
1, & \text{If a monitoring node is placed on } l_i, \forall l_i \in L \\
0, & \text{Otherwise} 
\end{cases} \quad (4.6)$$

Thus, the Linear Programming (LP) formulation of the monitoring problem can be formulated as follows:

$$\text{minimize } Z = |W|$$

subject to:

$$\omega_i \geq 1, \forall i, \ 1 \leq i \leq n, \quad (4.7)$$

$$l_i \leq 1, \forall i, \ 1 \leq i \leq x.$$ 

Now, we can review the method of the GA to solve the problem of static monitoring placement. According to the flowchart in Figure 4.3, we can determine the positions of the monitoring nodes as well as minimize the number of them by using the GA. The proposed technique, like other existing GA-based approaches [62], includes chromosome representation, initial population generation, fitness function definition, crossover, and mutation operations. In the subsequent subsections, we explain this algorithm and its steps.
Figure 4.3: Flowchart for GA.

- The location of Wireless Nodes
- The pre-defined location of Monitoring Nodes
- The Transmission Range of Wireless Nodes
- Objective: Minimizing the number of monitoring nodes
- Constraint: Each target node must be monitored by at least one monitoring node
- Pre-defined maximum number of iteration → #iteration = MaxIt.

Generating randomly binary chromosomes with length of the number of pre-defined location of Monitoring nodes (initial population)

Selecting best Chromosomes based on a Fitness function and Roulette-wheel selection method as parents

Crossover and Mutation of Chromosomes (Mutants and children population)

Merge (parents, children and mutants), sort (based on fitness function value) and truncate to create new generation

*NO* #iteration ≤ MaxIt.

*YES*

Best Chromosome = Optimal Solution for the problem

End
Chromosome Representation and Initial Population:

We define the chromosome for the GA as a binary representation (a string of zeros and ones). The length of each chromosome is equal to the number of pre-defined static monitoring locations. As shown in Figure 4.4, if the value of the i-th gene is 1, if a monitoring node is placed on the i-th pre-defined position. However, if the gene value is 0, no monitoring nodes are placed at the i-th potential location. For the initial population, the GA algorithm generates several chromosomes randomly based on the defined chromosome representation.

![Figure 4.4: Chromosome representation.](image)

Fitness Function:

A chromosome’s fitness value shows its ability to monitor all targets while using the fewest possible number of static monitoring nodes [62]. Thus, we define the fitness function as:

\[
\text{minimize } Z = |W| 
\]  

(4.8)
Selection Step:

In this step, the GA selects the best parents based on fitness values. The chromosomes with higher fitness values have more chance of being selected as parents. For solving the static monitoring placement problem, the GA selects some valid chromosomes with higher fitness values based on Roulette-wheel selection [62].

Crossover:

We use single-point crossover for the crossover process; a point is chosen at the random place of genes, and then applying a single-point crossover process as shown in Figure 4.5.

![Figure 4.5: Single point crossover.](image)

Mutation:

Some of the chromosomes undergo a mutation process applied at a randomly selected gene position where we randomly change its value to 0.
An Example:

We implement the GA algorithm in the MATLAB environment. As shown in Figure 4.6a and Figure 4.6b, 200 wireless sensor nodes (shown by circles) are randomly scattered in an area \((100m \times 100m)\). There are 66 pre-defined locations for static monitoring nodes considered in the area (shown by square). If we apply the GA algorithm, in iteration=1 (Figure 4.6b), we can see that 24 pre-defined locations are selected for deploying static monitoring nodes (shown by yellow square). In contrast, after 100 iterations, the algorithm selects 4 locations for deploying static monitoring nodes. In both Figure 4.6a and Figure 4.6b, you can see which sensor is assigned to which monitoring nodes that are away one-hop from target nodes. Therefore, we can apply the GA to different topologies of WSNs (like grid, random, etc.) to find the location of deploying a minimum number of static monitoring nodes.

4.7 UAV-based Monitoring

Another approach to measure RTT of wireless sensor nodes in tactical environments is using UAVs-based monitoring system. UAVs are aircraft that do not have human pilots. Drones are the common name for them. The use of UAVs has risen dramatically in recent decades. Drones have been used by the military since the 1980s [66]. Drones for commercial and amateur applications, on the other hand, have been increasingly popular in the last decade.
In this application, we need to use UAVs to measure RTT of each node and report them to the Monitoring-GW. As shown in Figure 4.7, we assume that the UAV is equipped with an out-of-band radio as well as an IEEE 802.15.4 radio interface. We assume the communication range of the UAV to connect the terrestrial sensor nodes is circular with a radius $R$. The UAV uses an omnidirectional antenna. In other words, the circular coverage area of UAVs is considered. Therefore, we need path planning for the UAV.
4.7.1 Path Planning for the UAV

We assume a grid of WSN with $m \times m$ in which $m$ is an even number, we can use the Hamiltonian path for path planning of the UAV [67]. Figure 4.8 shows an example of a WSN with 36 nodes. Using a Hamiltonian path allows the UAV to back to the charging station after the mission without extra distance flying. If we assume that the distance between each node is equal to $D$, we can calculate the total path of a flight:

$$\text{Total Path of a flight} = (4 \times (m - 1) + (m - 2)^2) \times D = m \times m \times D. \quad (4.9)$$

Now, we have to calculate the speed of the UAV. It should send $ICMPv6\_ECHO\_REQUEST$ and wait to receive the reply ($ICMPv6\_ECHO\_REPLY$) from the target.
nodes when it is in range of the target node(i). Please see Figure 4.7.

We can assume the worst case for response time ($T_R$) of each node and then calculate the speed of the UAV. The $T_R$ depends on the size of the network due to the use of the Carrier Sense Multiple Access (CSMA) protocol. Thus, the limitation speed of the UAV should be:

\[
\text{Speed of the UAV} \leq \frac{2R}{T_R}. \tag{4.10}
\]

According to the following formula, if we consider the maximum speed of the
UAV, we can calculate total flying time:

\[
\text{Total Flying Time} = \frac{\text{Total Path of a flight}}{\frac{2R}{T_R}}.
\]  

(4.11)

Regarding the maximum speed limit of commercial UAVs, some available models in the market, like DJI-FPV [68] (see Figure 4.9), can reach up to 39\(m/s\). Moreover, some fixed-wing UAVs are able to fly with maximum speeds varying from 17\(m/s\) to 293\(m/s\) [69].

![Figure 4.9: DJI-FPV can fly with maximum speed of 39\(m/s\) [68].](image)
4.8 Neighborhood Monitoring

Another alternative monitoring method for measuring RTT in WSNs is using the in-band monitoring method. In this approach, each node in a WSN is responsible for measuring RTT of its neighbors and transferring the measurements to the Monitoring-GW. This approach has its special advantages and disadvantages. The advantages include no need to deploy a parallel monitoring system and reducing the Total Cost of Ownership (TCO) of the network. On the other hand, the disadvantages are the risk of manipulating measurements by colluding or malicious nodes and some network overhead regarding bandwidth and energy of nodes because of the extra process of measurements and transferring them to the Monitoring-GW.

4.9 Results and Analysis

In this section, we examine some scenarios to evaluate the proposed method. We use the Cooja environment and parameters shown in Table 3.1 in section 3.6 except the number of nodes and the area that we explain in the subsections. The MATLAB environment is used to implement the detection engine module, as in the previous chapter. In addition, for simulation of UAV movement, we use an available Cooja plugin for node mobility [70].
Figure 4.10: A grid WSN with 64 nodes, 16 active eavesdroppers in area of $200m \times 200m$ in the Cooja environment.

**Scenario 1: using static monitoring nodes**

In this scenario, we consider a grid WSN with 64 nodes, and 25 percent of them are active eavesdroppers. The area is $200m \times 200m$. The locations of 9 static monitoring nodes are determined by the GA. In Figure 4.10, you can see the topology in the Cooja environment (yellow circles are normal nodes and purple circles are active eavesdroppers). The static monitoring nodes use `Ping6` command to measure the RTT of target nodes in the one-hop distance.

As Figure 4.11 shown, in this approach with measuring RTT, AED system can reach to $DR = 100\%$ and $AUC = 0.94$. The important point is that the AED system
using RTT is able to detect active eavesdroppers located at the border in contrast to the AED with intra-node delay measurements, which cannot detect them.

Scenario 2: using UAV-based monitoring

In this scenario, we consider the same topology as scenario 1. In order to examine the UAV-based monitoring system, we define a Hamiltonian path for a mobile node at the height of 5m with two different speeds (20m/s and 30m/s). To calculate the speed of the UAV, we use equation 4.10. In that formula, the $TR$ is the worst case of the response time of nodes in the network, so it can be determined by experiments. It
depends on the density of nodes in the network due to using IEEE 802.15.4 standard. In this case, the range of coverage of the UAV is $2R = 60m$. If we set $TR = 2s$, then we have the maximum speed of the UAV is $30m/s$, which also meets the speed limitation of commercial UAVs. In addition, we can calculate the total path of every round of flying the UAV by equation 4.9. It is equal to $1600m$ in this scenario. Moreover, the number of rounds that the UAV makes with $30m/s$ is about 67 rounds, and with $30m/s$ is 45 rounds during 60 minutes.

As Figure 4.12 and Figure 4.13 shown, the system with the UAV with the speed of $20m/s$ shows better performance ($AUC = 0.85$ in comparison to $AUC = 0.80$) than the system with the UAV with the speed of $30m/s$ because the UAV with higher speed misses measuring RTT for some nodes accurately as it moves fast and does not wait to get a response from them.

**Scenario 3: using neighborhood monitoring**

The identical topology as in scenario 1 is used in this scenario. To measure RTT, each node in the network sends ICMPV6_ECHO_REQUEST to its IPv6 neighbors and measures RTT. We assume that these measurements are sent to the Monitoring-GW in the same radio channel of communications between the nodes because they do not have any out-of-band radio interface. After that, the measurements are fed to the detection engine module of the AED system. As you can see in Figure 4.13,
Figure 4.12: ROC for the AED system with UAV-based monitoring with 20m/s speed.

The simulation results show $DR \geq 90\%$, $FPR \approx 0\%$, and $AUC = 0.96$ for this scenario. It is an excellent performance for the AED system. The reason is that the neighborhood monitoring system provides more holistic measurements of RTT for all nodes in the network, as each node is monitored by several neighborhoods. However, we need to analyze this monitoring approach’s cost, advantages, and disadvantages.
Figure 4.13: ROC for the AED system with UAV-based monitoring with 30m/s speed.

**Power Consumption and Overhead Comparison**

In this section, we analyze and compare the simulated network of the above scenarios in terms of power consumption and network overhead. According to the results, the highest performance belongs to the AED system with a neighbor monitoring system to measure RTT. As a result, we must compare them regarding energy consumption and network overhead to determine which solution best suits each network’s needs.

There are three key performance metrics for WSNs: average end-to-end delay,
average throughput, and average power consumption. We analyze the performance metrics for the simulated network of three different monitoring systems and also the network without any monitoring system.

Figure 4.15 shows a comparison between average end-to-end delay for three different systems and the network without a monitoring system as a baseline. As you can see, the end-to-end delay for the neighborhood monitoring system increases in comparison with other approaches because nodes have to perform the extra process to measure RTT and also send them to the GW and subsequently to the Monitoring-GW through the same channel of forwarding data packets. As a result, it causes to
increase in delay for data packets in the network. However, end-to-end delay for the
two other approaches do not show any significant changes.

The average throughput for three different systems is compared to the network
without a monitoring system as a baseline in Figure 4.16. As can be seen, the neigh-
borhood monitoring system’s average throughput decreases compared to other tech-
niques because of the extra process and forwarding of the measurements. On the
other hand, other techniques do not significantly impact the average throughput due
to their out-of-band operation to transfer the measurements.

Figure 4.17 displays the average power consumption of three distinct systems.
Using the neighborhood monitoring method causes higher average power consumption for nodes. As a result, it impacts the longevity of the network. Moreover, two other methods have little effect on this parameter. Therefore, using static monitoring nodes or UAV-based monitoring is more suitable for WSNs in inaccessible areas like tactical environments.

It is noteworthy to mention that, in addition to the performance metrics, we need to consider two important aspects regarding deploying neighborhood monitoring in tactical environments. Firstly, the risk of modification of measurements for nodes exposed in open environments is high because the measurements have to be transferred
through the same channel of packet forwarding, and malicious nodes may modify them to hide their eavesdropping activity. Thus, it is not robust to the falsification of malicious or colluding nodes. Moreover, deploying this method in already existing WSNs is also a challenging task due to changes in their running codes and protocols.

4.10 Conclusion

In this chapter, we proposed to use RTT measurement instead of intra-node delay to detect active eavesdroppers in WSNs. In order to measure RTT in a WSN, we can use three different approaches: static monitoring nodes, UAV-based monitoring,
and neighborhood monitoring. For placement of the static monitoring nodes, we formulate the problem and use the genetic algorithm as a meta-heuristic solution. In addition, we discussed a UAV-based monitoring system regarding path planning and speed. Another alternative way to measure RTT is using the neighborhood monitoring method. All of these approaches showed good performance regarding $AUC$ and also $DR$ through simulation scenarios. In contrast to using intra-node delays measurements, it is important to note that using RTT as an input feature can help the AED system to detect active eavesdroppers that do not forward packets of their neighbors for some reason.
Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this thesis, we proposed a novel system for detecting the presence of active eavesdroppers in homogeneous multi-hop WSNs. The proposed system consists of two essential components: monitoring and detection engine modules. The monitoring module uses an out-of-band approach to measure intra-node delays of nodes as a key feature for detecting active eavesdroppers in WSNs. This module can utilize asynchronous static monitoring nodes or UAVs for areas with low accessibility in tactical environments. In addition, the proposed architecture can be deployed easily for already established WSNs without any modification, and it is robust to the falsification of malicious nodes. The Z-test anomaly detection algorithm is used in the detection engine module. It is a lightweight algorithm for running on edge devices. Simulation results showed that using intra-node delays of nodes as an input feature
for the AED system has some drawbacks. For instance, active eavesdroppers who do not relay packets because of their locations or other reasons cannot be detected by the AED system. Therefore, we proposed to use RTT measurements. In this method, the monitoring module sends requests to get responses from nodes, and based on the response delay, the AED can detect active eavesdroppers in WSNs. For using this measurement, we reviewed three different monitoring systems, namely static monitoring nodes, UAV-based monitoring, and neighborhood monitoring. The simulation results showed that the AED system using RTT is able to detect active eavesdroppers, regardless of their locations, with a high detection rate (100%) and a low false-positive rate ($\leq 5\%$) with excellent performance ($AUC \approx 0.97$). In addition, we analyzed the overhead of the in-band monitoring system (neighborhood monitoring) in terms of network performances.

### 5.2 Future Work

For future work, there are the following directions to expand and enhance this thesis:

- Adding another related feature like the temperature of nodes or energy consumption parameters.

- Using the Infrared thermography method to measure the temperature of nodes as a related feature.
• Deploy the AED system on a testbed to get real-world results.

• Developing a mathematical model for the intra-node delay to backup our simulation results.
References


