Secure Data Collection in Wireless Sensor Networks

Wael Y. Alghamdi

A thesis in fulfillment of the requirements for the degree of

Doctor of Philosophy

School of Computer Science and Engineering

Faculty of Engineering

The University of New South Wales

September 2017

THE UNIVERSITY OF NEW SOUTH WALES
Thesis/Dissertation Sheet

Surname or Family name: **Alghamdi**

First name: **Wael Y.** Other name/s:

Abreviation for degree as given in the University calendar: **PhD**

School: **School of Computer Science and Engineering** Faculty: **Faculty of Engineering**

Title: Secure Data Collection in Wireless Sensor Networks

Abstract 350 words maximum

Sensor nodes have limited processing power, small storage capacity and limited energy. These constraints make classical security algorithms unsuitable for WSNs (Wireless Sensor Networks). Therefore, new techniques that consider these limitations are needed. WSNs have a wide range of applications, including military field surveillance, healthcare, homeland security, industrial control, and intelligent green aircraft. Therefore, network security has become increasingly important. There are various types of attacks that may cause security problems, such as modification attacks and selective forwarding attacks.

This thesis investigates three security problems in WSNs. Firstly, we investigate the problem of minimizing the failure rate of packet delivery in the presence of modification attacks and selective forwarding attacks in a static WSN with one base station without using expensive encryption/decryption algorithms. We propose a novel heuristic approach to this problem. Our approach is based on randomized multipath routing.

Secondly, we investigate the problem of constructing a shortest path overhearing tree with the maximum lifetime for data collection. We propose three approaches for homogeneous WSNs and heterogeneous WSNs. The first one is a polynomial-time heuristic approach. The second one uses ILP (Integer Linear Programming) to iteratively find a monitoring node and a parent for each sensor node. The last one optimally solves the problem by using MINLP (Mixed- Integer Non-Linear Programming).

Lastly, we investigate the reliable and secure end-to-end data aggregation problem considering selective forwarding attacks and modification attacks in homogeneous cluster-based WSNs, and propose three data aggregation approaches which can defend against both modification attacks and selective forwarding attacks. Our approaches use secret sharing and signatures to allow aggregators to aggregate the data without understanding the contents of messages and the base station to verify the aggregated data and retrieve the raw data from the aggregated data.

FOR OFFICE USE ONLY Date of completion of requirements for Award

Originality Statement

'I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.'

Wael Y. Alghamdi
September 14, 2017

Copyright Statement

_____

Wael Y. Alghamdi
September 14, 2017

Authenticity Statement

_____

Wael Y. Alghamdi
September 14, 2017

Dedication

I dedicate this thesis to my wife, **Bayan Alsharbi**, my children, **Wesam and Talleen**, and **My Parents**.

Acknowledgment

Thankfulness to my wife, ***Bayan Alsharbi***, whose love and encouragement allowed me to finish this thesis. She already has my heart so I will just give her a heartfelt thanks.

Many thanks to my supervisor at The University of New South Wales ***Dr.Hui Wu*** who constantly gives me precious suggestions and guidance during my research and studies at the university. He spend a great amount of time helping and supporting me. He also gave me useful suggestions and guidelines to facilitate a good quality of research and publications.

Also, many thanks for my Co-Supervisor ***Dr.Salil S.Kanhere***, for his time, comments, and advice during my PhD.

Many thanks to ***my colleagues*** and research group in The University of New South Wales. We shared everything among several years, supported each other, and contributed with each other towards a successful research.

Thanks to my children, ***Wesam*** and ***Talleen*** who came to my life with brightness. You have made me stronger, better and more fulfilled than I could have ever imagined.

Thankfulness to ***My Parents*** who raised me up and took care of me throughout my life and to my brothers for all the good times that we had together.

Gratefulness to my mother-in-low, ***Prof. Samirah Kurdi*** for her endless support and motivations.

Finally, the ***Ministry of Higher Education*** in Saudi Arabia along with the **Saudi Arabian Cultural Mission** in Australia for their support during the period of my scholarship, which helped me achieve my goals toward a successful career.

Publication

- Wael Y. Alghamdi, Hui Wu, Fei Jingjing, Salil Kanhere, Randomized Multipath Routing for Secure Data Collection, The IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (IEEE ISSNIP 2014 - Security Privacy and Trust for Cyber-Physical Systems).
- Jingjing Fei, Hui Wu, Wael Y. Alghamdi. Lifetime and Latency Aware Data Collection Based on k-Tree. The IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2015.
- Wael Y. Alghamdi, Hui Wu, Wenguang Zheng, Salil S. Kanhere. Constructing A Shortest Path Overhearing Tree with Maximum Lifetime In WSNs. IEEE-Hawaii International Conference on System Sciences (HICSS), 2016.
- Wael Y. Alghamdi, Hui Wu, Salil S. Kanhere. Reliable and Secure End-to-End Data Aggregation Using Secret Sharing in WSNs. IEEE Wireless Communications and Networking Conference (WCNC), 2017.

Abstract

Sensor nodes have limited processing power, small storage and limited energy. These constraints make classical security algorithms unsuitable for WSNs (Wireless Sensor Networks). Therefore, new techniques considering these limitations are needed. WSN has a wide range of applications, including military field surveillance, health-care, homeland security, industrial control, and intelligent green aircraft. Therefore, network security becomes increasingly important. There are various attacks that may cause many security problems such as the modification attack and the selective forwarding attack. In a modification attack, a malicious sensor node modifies a packet it receives and sends the incorrect packet to the base station via a routing path. In a selective forwarding attack, a malicious sensor node may refuse to forward a packet, resulting in packet loss.

This thesis investigates three security problems in WSNs. Firstly, we investigate the problem of minimizing the failure rate of packet delivery in the presence of modification attacks and selective forwarding attacks in a static WSN with one base station without using expensive encryption/decryption algorithms. We propose a novel heuristic approach to this problem. Our approach is based on randomized multipath routing. When a sensor node needs to send a packet to the base station, it creates three copies and sends them to the base station via three paths. Of the three paths, two of them are selected at random based on a spanning tree with the base station as the root. The base station accepts a packet only if it receives at least two identical copies. We have simulated our approach and compared it with the state-of-the-art approach. The simulation results show that our approach achieves a very low failure rate of packet delivery in the presence of a relatively high percentage of malicious sensor nodes. Our approach provides an average failure rate improvement of 72.9%, an average network lifetime improvement of 84.5%, and an average latency improvement of 30.9% over the state-of-the-art approach.

Secondly, we investigate the problem of constructing a shortest path overhearing tree with maximum lifetime for data collection. We propose three approaches for homogeneous WSNs and heterogeneous WSNs. The first one is a polynomial-time heuristic. The second one uses ILP (Integer Linear Programming) to iteratively find a monitoring node and a parent for each sensor node. The last one optimally solves the problem by using MINLP (Mixed-Integer Non-Linear Programming). We have implemented the three approaches using MIDACO solver and MATLAB Intlinprog, and performed extensive simulations using NS2.35. In homogeneous networks, the simulation results show that the average

lifetime of all the network instances achieved by the heuristic approach is 85.69% of that achieved by the ILP-based approach and 81.05% of that obtained by the MINLP-based approach, and the performance of the ILP-based approach is almost equivalent to that of the MINLP-based approach. In heterogeneous networks, the simulation results show that the average lifetime of all the network instances achieved by the heuristic approach is 87.371% of that achieved by the ILP-based approach and 74.9% of that obtained by the MINLP-based approach, and the performance of the ILP-based approach is almost equivalent to that of the MINLP-based approach. One limitation of our shortest-path overhearing tree is that if a monitor colludes with the parent of the monitored sensor node, the attacks by the parent of the monitored sensor node may not be detected. We overcome this problem by selecting a different monitor for every sensor node periodically.

Lastly, we investigate the reliable and secure end-to-end data aggregation problem considering selective forwarding attacks and modification attacks in homogeneous cluster-based WSNs, and propose three data aggregation approaches. Our approaches, namely, Sign-Share and Sham-Share, use secret sharing and signatures to allow aggregators to aggregate the data without understanding the contents of messages, and use the base station to verify the aggregated data and retrieve the raw data from the aggregated data. We also modify Sign-Share to allow malicious node detection in a multi-hop structure. To best of our knowledge, this is the first lightweight en-routing malicious node detection in concealed data aggregation. We have performed extensive simulation to compare our approaches and the two state-of-the-art approaches, PIP and RCDA-HOMO. The simulation results show both Sign-Share and Sham-Share consume a reasonable amount of time in processing the data and aggregating the data. The simulation results also show that our first approach achieves an average network lifetime of 102.33%, and an average aggregation energy consumption of 74.93% over PIP. Also, it achieves an average aggregation processing time and sensor data processing time of 95.4%, 90.34% over PIP, and 98.7%,92.07% over RCDA-HOMO, respectively, while it achieves an average network delay of 71.95% over PIP.

Keywords

Overhearing, *Multipath Routing*, *Integer Linear Programming*,*Mixed- Integer Non-Linear Programming*, *Aggregation*, *Digital Signature*, *Modification Attacks*, *Selective Forwarding Attacks*, *Heuristic*, *Polynomial-Time*, *Secret Sharing*

Contents

List of Figures

Chapter 1

Introduction

This chapter describes the significance of the problems investigated in this thesis and some background information about wireless sensor networks, attacks and security requirements. Furthermore, it outlines the major contributions of our work, and provides the thesis structure.

1.1 Introduction

A Wireless Sensor Network (WSN) consists of a large number of autonomous sensors (motes) and a base station (sink). All sensors are responsible for sensing and collecting certain information from the surrounding physical environment, as shown in Figure 1.1. WSNs have become very popular in recent years due to their low cost and their ability to solve and interact with so many real-world problems and challenges.

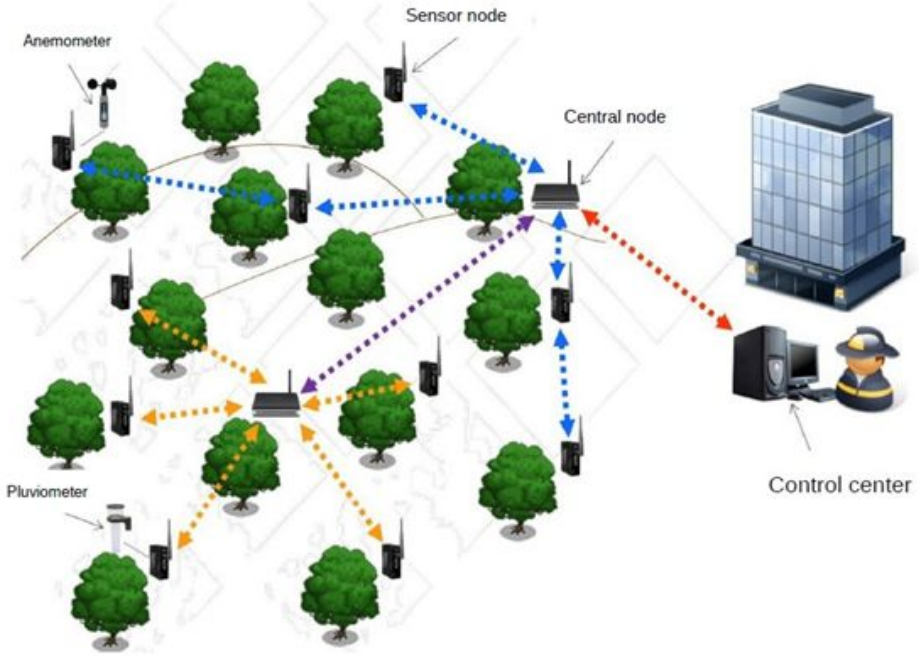

Figure 1.1: Wireless Sensor Network

Lately, the number and variety of smart environments have significantly increased [1, 2] and WSNs are involved in a wide range of applications, including military field surveillance, healthcare, homeland security, industrial control, and intelligent green aircraft [1–3]. Therefore, network security has become increasingly important.

This thesis is focused on secure data collection in wireless sensor networks. Collecting data securely is challenging in the presence of network constraints and limitations. Various attacks may occur, which may result in data loss, false data, and private data known to attackers. As a result, efficient algorithms for secure data collection need to be proposed and the algorithms need to consider the security requirements and various constraints on WSNs.

1.2 Background

In this section, firstly, we discuss in detail the constraints and limitations of WSNs. Secondly, we discuss security requirements. Lastly, we overview the major attacks in WSNs.

1.2.1 Wireless Sensor Network Constraints and Limitations

There are many constraints and limitations involved in wireless sensor networks, as shown in Figure 1.2. It is difficult to implement existing security mechanisms for other networks in a WSN due to these limitations and constraints. Accordingly, a secure WSN requires different approaches that satisfy the constraints [3–6]. WSN constraints and limitations are detailed as follows:

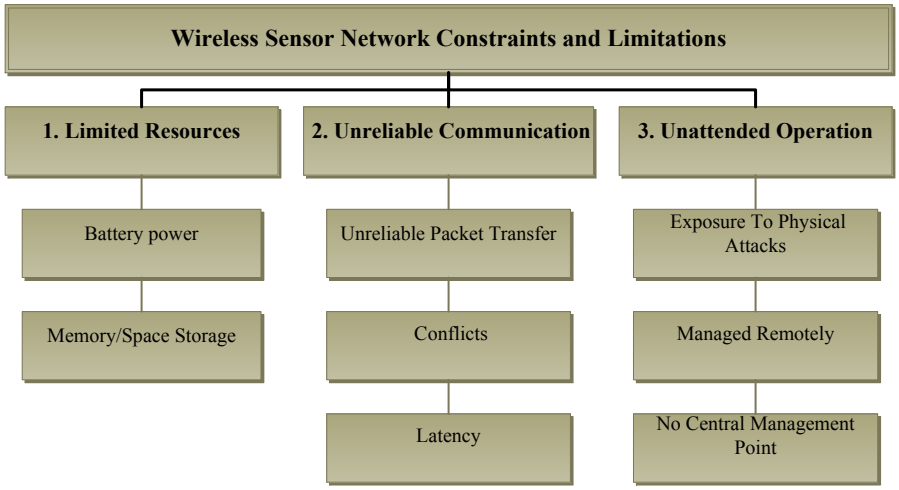

Figure 1.2: WSN Constraints and Limitations

1. Limited Resources: any security mechanism for WSNs must take the limited resources of WSNs into consideration. [3,5,7]. WSN resource limitations are on battery power, memory, processing power, and communication as described below:

- Low battery power: energy is the most important limitation in WSNs because depleting a sensors battery may cause a network disconnected. Furthermore, replacing or recharging the sensors battery is incredibly difficult due to the fact that WSNs are mostly deployed in hostile environments. The lifetime of a sensor node is completely dependent on the nature of the deployed protocols and algorithms. Therefore, the proposed security mechanism must ensure consideration of the sensors energy requirements [3, 5, 6].
- Small memory and low processing power: in order to reduce the manufacturing cost, the sensor nodes have small memory storage and low processing power [5, 6]. Any algorithms for WSNs need to have small code size, data size and be fast. For example, TelosB is one of the most common sensor nodes, which has an 8 MHz TI MSP430 micro-controller with 10KB RAM [3].

2. Unreliable Communication: communication is another challenge due to the broadcast nature of WSNs. Unreliable packet transfer, conflicts, and latency are inherent natures in WSNs.

   - Unreliable packet transfers: WSNs transfer packets between sensors and BS through wireless channels that are inherently unreliable. Unreliable communication is caused by channel errors or congestions and results in packet loss and corruption [3, 6].
   - Conflicts: often, the communication in WSNs is unreliable even with the existence of reliable channels. The conflict between signals with similar frequencies may cause data loss and corruption [3].
   - Latency: in general, is the time difference between generating the data at the source and receiving the data at the destination. It is affected by many factors such as network congestion, data processing time and routing in multi-hop [3,6].

3. Unattended Operations: sensor nodes deployed in an open environment will be left unattended for a long period of time. Accordingly, there are three challenges, namely,

physical attacks, remote management, and the lack of a central management point [3] as described below:

- Exposure to physical attacks: WSNs are subject to physical attacks in an open environment as they can be easily accessed by intruders and affected by factors like bad weather [4].
- Remote management: the physical damage to a sensor is hard to detect in a remotely managed wireless sensor network.
- No central management point: WSN is a distributed system where each sensor node is autonomous. A single sensor node failure may cause the whole network to stop functioning if the system is not resilient to sensor node failures. No central management point poses many challenges to the WSN design [3].

WSNs are becoming increasingly popular due to their low-cost and their effectiveness in collecting data across various environments. However, WSNs suffer from many constraints and limitations. One of them is that sensor nodes have limited power and energy. In this thesis, we aim at increasing security while maintaining low usage of battery power in order to improve the network lifetime.

1.2.2 Wireless Sensor Networks Security Requirements

A WSN is a special type of network that differs from a typical network due to the constraints and the characteristics of the network. Figure 1.3 shows the security requirements for WSNs, which combine the security requirements for typical networks and the unique requirements for WSNs. There are a number of security requirements identified through the literature. The previous research [3, 8–10] identifies a list of requirements shown as follows:

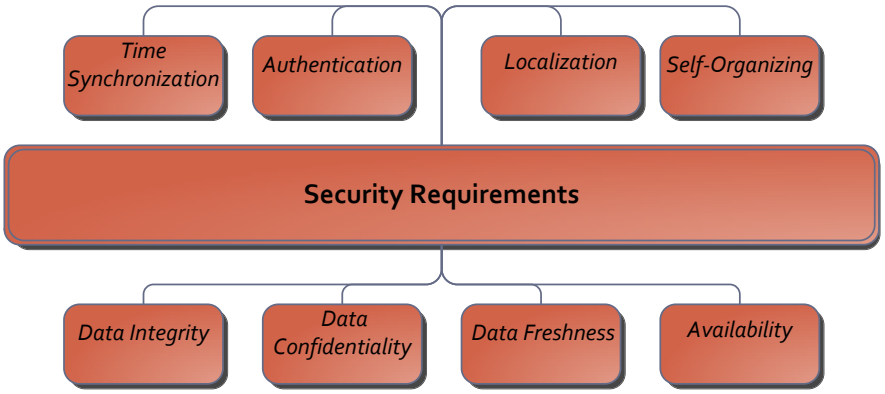

Figure 1.3: WSN Security Requirements

Data Confidentiality, Integrity, and Availability

- Data Confidentiality includes ensuring that unauthorized personnel cannot have access to the data. It is one of the most important requirements in network security and especially in WSNs. Every network concerned with data security takes proper measures to ensure data confidentiality, so that only authorized sensor nodes understand the content of a message [11]. This can be done in many ways such as encryption and secret sharing techniques [12]. WSNs have the three data confidentiality requirements [3, 10]: a) Raw data and sensor readings should be confidential and should not leak to any unauthorized party, especially in critical applications. b) There is a secure channel for sensitive information. c) Public information such as public keys and sensor identities should be kept highly encrypted [8, 9].

- Data Integrity is concerned with the accuracy and consistency of data [9–11]. The sensory information could be manipulated via an infected sensor node in the network that harms the network and causes disarray [3, 8, 13].

- Availability indicates that a sensor node remains alive and can produce data whenever needed [14]. There are a number of factors that can impact the availability

of sensor nodes and base stations. Firstly, added computation requires more energy, thus shortening the sensor lifetime. Secondly, unnecessary communication will cause additional energy consumption. Accordingly, the data will not be available for long [5]. Finally, some applications are based on the single point scheme, such as an aggregator in a cluster. A single point failure in such schemes renders data unavailable [3, 8].

Authentication

Finally, some applications are based on the single point scheme, such as an aggregator in a cluster. A single point failure in such schemes renders data unavailable [11]. Attackers are not only able to modify data packets, but are also capable of injecting false data to disarray the network. Therefore, authentication is the key to verifying the source of data so that the base station or the data destination is confident that the data has not been manipulated. The network design must ensure that there is a proper authentication for administrative tasks done on the network [3,8,10]. For example, the symmetric mechanism is used to achieve authentication for two parties. In this technique, a secret key is shared for authentication and transfer of data [3, 8].

Time and Location

There are two aspects for time requirements in WSN, including time synchronization and data freshness. The former refers to the WSN being modified to be time synchronized to exchange data, while the latter implies that the sensed data are recent, and ensures that no obsolete data are replayed [3, 11]. Otherwise, the network is susceptible to replay attacks [4]. Finally, localization in WSN often refers to the ability to accurately and automatically locate each sensor node connected to the network [3].

Self-Organization

Finally, sensor nodes in wireless network should have robustness and flexibility to self-organize to meet non-fixed ad-hoc structure requirements, particularly in difficult situations. This includes the ability to determine the routes using multi-hop routing conduct key management tasks and build trust [3].

There are a number of security requirements that need to be addressed in WSNs. This thesis, however, focuses on maintaining integrity and confidentiality of data-in-transit. This is due to the possible catastrophes a breach of such data might cause, particularly in mission critical applications. A security attack of data-in-transit in WSNs may mislead the base station to make inappropriate decisions.

1.2.3 Wireless Sensor Network Security Attacks

Due to the low cost of sensor nodes and their popularity, WSNs are widely used to detect physical activities in a certain environment, and have a wide range of applications. Thus, new techniques considering their limitations and security are required.

Various attacks can be performed against WSNs, as shown in Figure 1.4. Generally, we classify attacks as Passive Attacks and Active Attacks. A Passive Attack is the one in which the target is monitored and scanned for vulnerabilities. The purpose is solely to gain information about the target and no data is changed on the target. An Active Attack is the one in which the attacker attempts to make changes to data on the target or data en-route to the target.

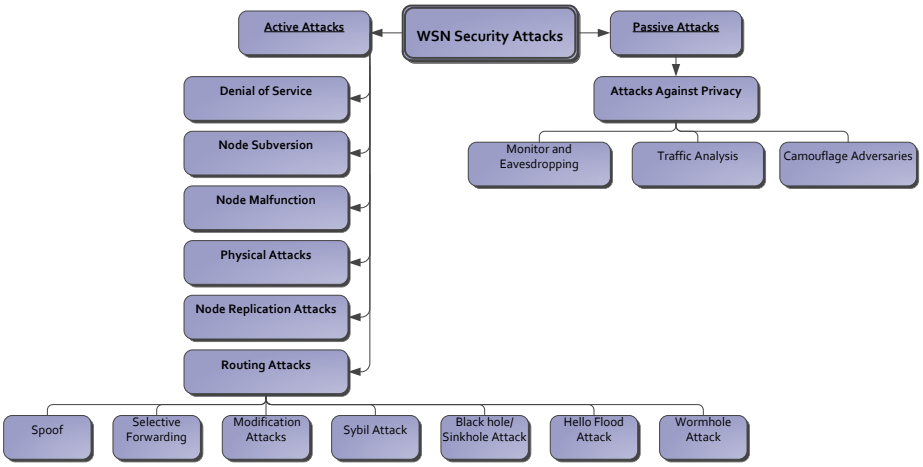

Figure 1.4: WSN Security Attacks Classification

Passive Attacks

After a long period of listening to wireless channels in a WSN, the attackers may understand the network and protocol behaviour. They can then perform various attacks such as attacks against the network's privacy, traffic analysis attacks, camouflage of adversaries attacks, and monitor/eavesdropping attacks [9].

1. Attacks Against Privacy:

   The wireless sensor network is capable of producing a huge amount of data after deployment. Although sensor networks offer great benefits to their users, they are susceptible to abuses of privacy [3]. This is shown by the famous example of the panda-hunter problem [15], in which the hunter identifies the position of a panda after monitoring and tracking the data traffic.

   (a) **Monitor and Eavesdropping**

   One of the most obvious attacks against privacy occurs when the attacker intends to listen to data transmission over the wireless channels to determine the

communication contents of the network. Accordingly, if the data contains details about the network configuration, eavesdropping will expose the network's privacy [3].

(b) **Traffic Analysis**

Traffic analysis is combined with the previous attacks to identify sensor nodes with registered activities. The main intention is to investigate the sensor nodes with special roles, such as aggregators, via traffic analysis [3].

(c) **Camouflage Adversaries**

Attackers perform camouflage by inserting a node into a network for a period of time until it appears to be a normal node to its neighbours. Once this goal is achieved, it starts to misroute packets in the network [3].

Active Attacks

Active Attacks intend to inject the network with false information, and/or modify the collected data, to create holes in the security protocols against the data or destroy the network and shorten its lifetime [2, 9].

1. Routing Attacks:

(a) **Spoof, Altered and Replayed Routing Information**

Attackers performing spoofing attacks masquerade as another node, and therefore gain an illegitimate advantage. Eventually, this creates routing loops, extends or shortens a routing path, initiates false information, and partitions the network [2].

(b) **Selective Forwarding**

In the selective forwarding attacks, a malicious sensor node may deliberately drop some packets received from other sensor nodes, resulting in packet loss [2, 16].

(c) **Message Corruption/Modification Attack**

A modification attack is very dangerous for the network. A malicious sensor node may modify some packets received from other sensor nodes and forward the incorrect packets to the base station [16].

(d) **Sybil Attack**

This attack is a malware device that takes a number of identities. In a peer-to-peer network, a Sybil attack defeats the redundancy mechanism within the distributed storage system that contains data. Unlike a node replication attack, a single node appears with multiple identities [2]. This attack is effective on data aggregation routing algorithms and voting, foiling misbehaviour detection, and resource allocation. Mainly, it intends to degrade the data integrity and acquire a disproportionate level of control over the network [3, 5].

(e) **Black Hole/ Sinkhole Attack**

A malicious node is introduced into the network and made to act as black hole to attract nodes in the network, and then discards packets [2]. This node attracts all data traffic and packets throughout the network. For instance, in the flooding protocol the message in the wireless sensor network takes the shortest path to reach the destination. The sinkhole nodes makes changes to these paths, brings all the messages to itself and destroys them [5, 17] .

(f) **Hello Flood Attack**

The major weapon of such an attack is HELLO packets sent to many sensor nodes with high radio transmission. This is done to convince sensor nodes that the attacker is their neighbour. Accordingly, the victim nodes send their sensitive data through the attacker [5]. The main goal of the attacker is to drain the network energy [2].

(g) **Wormhole Attack**

In this critical attack, the malicious node records packets at one location in the network for a period of time. Then, it tunnels those packets to another

location which increases the processing time and congestion in the network, thus consuming more energy and shortening the network lifetime [5].

2. Denial of Service

   This attack jams sensor nodes in the wireless sensor network using different strategies [2]. The jamming refers to the radio signal that interferes with the original signal of sensor nodes and stops normal communication. The jamming is found to be in two forms: intermittent jamming and constant jamming.

   Intermittent jamming makes the node transmit a message periodically over time to make the transmission noticeably slow [4]. On the other hand, constant jamming makes all messages impossible to transmit and blocks all the nodes. This attack completely shuts down all communication in the network [3].

   The other way to deny service is to attack the link layer of the network, which violates the IEEE 801.11b protocol completely. The message is sent constantly by the attacker to generate collision, which requires retransmission of affected packets. When the retransmission is too high, the power supply of sensor nodes may be disrupted.

3. Node Subversion

   In this attack, the main intention is to capture one node to obtain its cryptographic keys, thereby compromising the entire network [5].

4. Node Malfunction

   In this attack, the malicious node intends to generate false information and inject it into the network to mislead the network and prevent it from making right decisions [5].

5. Physical Attacks

   By nature, sensor nodes are deployed in open environments that are found to be more vulnerable to physical attacks. The major goal of the attacker is to destroy a

sensor node to serve a different goal, such as partitioning the network, and to destroy sensitive information collected by that sensor node [3, 5] .

6. Node Replication Attacks

   In a node replication attack, the attacker intends to insert a new node into the network by copying the ID of one of the existing nodes. In this attack, multiple nodes appear to have a similar identity [2]. Therefore, the performance of the network is seriously disturbed [13]. The attacker may perform packet misrouting or even corruption. The connection of the network with the malicious sensor is disrupted and causes network partitioning [5].

As shown above, with the increasing number of applications in WSN, modification attacks and selective forwarding attacks, remain a major concern that endangering the integrity and confidentiality of data-in-transit. WSNs often include mission critical applications such as military surveillance, healthcare, national security, industrial control, and intelligent green aircrafts. Exposing data-in-transit in mission critical applications or interfering in its process may lead to unforeseeable consequences [2,3,6]. Thus, this thesis aims to propose methods that can support the integrity and confidentiality of WSNs. Mission critical applications are the main target applications of the proposed methods. The next section 1.3 highlights and identifies a number of key research issues and major contributions.

1.3 Key Research Issues and Major Contributions

In this thesis, we investigate the following three security problems:

1. The first problem is to minimize the failure rate of packet delivery in the presence of modification attacks and selective forwarding attacks in a static WSN with one base station without using expensive encryption/decryption algorithms.

2. The second problem is to construct a shortest path overhearing tree with the maximum lifetime for a homogeneous network or a heterogeneous network.

3. The third problem is reliable and secure end-to-end data aggregation considering selective forwarding attacks and modification attacks in homogeneous cluster-based WSNs.

We make the following major contributions.

- Firstly, we propose a novel heuristic approach to the first problem. Our approach is based on randomized multipath routing. When a sensor node needs to send a packet to the base station, it creates three copies and sends them to the base station via three paths. Of the three paths, two of them are selected at random based on a spanning tree with the base station as the root. The base station accepts a packet only if it receives at least two identical copies. We have simulated our approach and compare it with the most relevant randomized algorithm. The simulation results show that our approach achieves a very low failure rate of packet delivery in the presence of a relatively high percentage of malicious sensor nodes. Our approach provides a failure rate improvement over the most relevant approach by 72.9% on average, and the lifetime of the network is improved by 84.5% while the latency is improved by 30.9%.

- Secondly, we propose three approaches to the second problem. The first one is a polynomial-time heuristic. The second one uses ILP (Integer Linear Programming) to iteratively find a monitoring node and a parent for each sensor node. The last one optimally solves the problem by using MINLP (Mixed-Integer Non-Linear Programming). We have implemented the three approaches using MIDACO solver and MATLAB Intlinprog, and performed extensive simulations using NS2.35. In homogeneous networks, the simulation results show that the average lifetime of all the network instances achieved by the heuristic approach is 85.69% of that achieved

by the ILP-based approach and 81.05% of that obtained by the MINLP-based approach, and the performance of the ILP-based approach is almost equivalent to that of the MINLP-based approach. In heterogeneous networks, the simulation results show that the average lifetime of all the network instances achieved by the heuristic approach is 87.371% of that achieved by the ILP-based approach and 74.9% of that obtained by the MINLP-based approach, and the performance of the ILP-based approach is almost equivalent to that of the MINLP-based approach. One limitation with our shortest path overhearing tree is that if a monitor colludes with the parent of the monitored sensor node, the attacks by the parent of the monitored sensor node may not be detected. We overcome this by selecting a different monitor for every sensor node periodically.

- Lastly, we propose two data aggregation approaches to the third problem. Our approaches, Sign-Share and Sham-Share, use secret sharing and signatures to allow aggregators to aggregate the data without understanding the content of messages and allow the base station to verify the aggregated data and retrieve the raw data from the aggregated data. We also modify Sign-Share to allow en-routing malicious node detection in a multi-hop structure. We have performed extensive simulations to compare our approaches with the two state-of-the-art approaches, PIP and RCDA-HOMO. The simulation results show that both Sign-Share and Sham-Share consume a reasonable amount of time in processing the data and aggregating the data. The simulation results show that our first approach achieved an average network lifetime of 102.33% over PIP, and an average aggregation energy consumption of 74.93%. Also, it achieved an average aggregation processing time and sensor data processing time of 95.4% and 90.34%, respectively, over PIP, and 98.7% and 92.07%, respectively, over RCDA-HOMO, while it achieved an average network delay of 71.95% over PIP. Although RCDA-HOMO is a completely different technique, a comparison is performed to measure the computational overhead.

- For each problem investigated in this thesis, we follow the following methodology.

Firstly, we perform a comprehensive literature review. Secondly, we construct a model for the problem. Thirdly, we propose approaches that solve the problem and/or heuristics that find an approximate solution. Fourthly, we validate our approaches by performing analyses and extensive simulations and by comparing them with the state-of-the-art approaches.

1.4 Dissertation Organization

This thesis is organized as follows:

Chapter 2 describes our approach to the first problem. In this chapter, we detail the problem, the related work, the network model, the proposed novel approach, and finally the results.

Chapter 3 describes our approaches to the second problem. In this chapter, we describe the problem, the related work, the network model, the proposed heuristic approach, ILP, MINLP, and finally the results.

Chapter 4 describes our approaches to the third problem. In this chapter, we explain the problem, the related work, the network model, the proposed three approaches, and finally the results.

Chapter 5 presents our conclusion and future works.

Chapter 2

Randomized Multipath Routing for Secure Data Collection

WSN (Wireless Sensor Network) has a wide range of applications. As a result, security problems become increasingly important. In this chapter, we investigate the problem of minimizing the failure rate of packet delivery in the presence of the modification attacks and the selective forwarding attacks in a static WSN with one base station without using expensive encryption/decryption algorithms. Firstly, we propose a novel heuristic approach to this problem. Our approach is based on randomized multipath routing. When a sensor node needs to send a packet to the base station, it creates three copies and sends them to the base station via three paths. Among the three paths, two of them are selected at random based on a spanning tree with the base station as the root. The base station accepts a packet only if it receives at least two identical copies. Secondly, we compare our approach with a state-of-the-art approach and show our simulation results.

2.1 Introduction

A WSN consists of a large number of autonomous sensor nodes and one or more base stations. Each sensor node sends its data sensed from the physical environment to its designated base station. Typically, sensor nodes are battery powered. In order to save energy, the power of transceiver of each sensor node is kept low, leading to a short transmission range. As a result, data collection is performed in a multi-hop way. Each packet originated from a sensor node needs to delivered to the target base station via a routing path. Different routing structures such as trees have been proposed [18].

WSN has a wide range of applications, including military field surveillance, health-care, homeland security, industrial control, and intelligent green aircraft [3]. Therefore, network security becomes increasingly important.

Sensor nodes have limited processing power, small storage and limited energy. These constraints make classical security algorithms unsuitable for WSNs. Therefore, new techniques considering these limitations are needed.

WSN security has attracted extensive researchers [3,5,13,19–22]. There are various attacks that may cause many security problems. Among them are the modification attack and the selective forwarding attack. In the modification attack, a malicious sensor node may modify a packet it receives and sends the incorrect packet to the base station via a routing path. In the selective forwarding attack, a malicious sensor node may refuse to forward a packet, resulting in a packet loss.

We investigate the problem of minimizing the failure rate of packet delivery in the presence of the modification attacks and the selective forwarding attacks in a static WSN with one base station without using expensive encryption/decryption algorithms. The failure rate is equal to the percentage of the total number of packets rejected by the base station over the total number of packets generated by all the sensor nodes. We propose a novel

heuristic approach to this problem. Our approach is based on the three-copy strategy. When a sensor node generates a packet, it creates three copies which are sent to the base station via three paths. Among the three paths, two of them are selected at random based on a spanning tree with the base station as the root. The base station accepts a packet only if it receives at least two identical copies.

We make the following key contributions:

- We propose a heuristic novel approach aiming at minimizing the failure rate of packet delivery in the presence of modification attacks and selective forwarding attacks. This approach consists of a distributed naming algorithm and a randomized multipath routing scheme.
- We have performed extensive simulations of our approach and compared it to with a state-of-the-art approach. The simulation results show that compared with the state-of-the-art approach, our approach achieves an average improvement of 72.89% in failure rate, an average improvement of 84.5% in lifetime, and an average improvement of 30.9% in latency, respectively.

The rest of this chapter is organized as follows. Section 2.2 discusses the WSN security attacks and the related work on multipath routing. Section 2.3 describes the proposed approach in detail. Section 2.4 shows the simulation results, analyses, and key observations. Lastly, Section 2.5 concludes this chapter.

2.2 Related Work

2.2.1 Modification Attack

Only a few attempts have been made to handle the modification attack so far. [23] uses an overhearing technique to detect malicious packet-modifying attacks in WSNs. By the overhearing technique, a committee structure is constructed for each sensor node. The committee structure includes several committee sets, and each committee set is designed for a specific communication link. Due to the microwave nature of the wireless channel, neighbouring sensor nodes within a sender's radio range can overhear the packet the sender is transmitting. Therefore, each packet can be examined by the sensor nodes of the committee set during forwarding. If a packet is modified by a malicious node, the committee set will detect the error. However, the overhearing technique consumes a significant amount of energy [24] unless the overhearing topology construction algorithm optimizes the network lifetime when constructing an overhearing topology as per shown in chapter 3. In the following chapter, we will discuss in details how construct a shortest path overhearing tree with the maximum lifetime.

[22] proposes an approach for identifying the malicious nodes that modify or drop packets. The proposed approach encrypts each packet and adds some extra bits to the packet to hide the source of the packet. It adds a packet mark, a small number of extra bits to each packet such that the base station can recover the source of the packet and figure out the dropping ratio associated with every sensor node. The routing tree structure dynamically changes in each round so that behaviour of each sensor node can be observed in a large variety of scenarios. The heuristic ranking algorithms can identify most of the bad nodes with small false positive. [22] also gives an excellent review of the previous approaches to the modification attack problem and the selective forwarding attack problem.

2.2.2 Node-Disjoint Multipath Routing

It is widely agreed that multipath routing is an efficient solution to the modification and selective forwarding attacks [13, 19, 20, 25]. Multipath routing reduces the chance of a packet being modified or dropped by a malicious sensor node by using different paths. A survey of multipath routing protocols is presented in [26]. Multipath routing can be either node disjoint [26], or link disjoint [26], or partially disjoint [27].

[28] presents a multipath protocol to increase the transmission reliability by discovering a back-up path beside the service-path in case of transmission failures. [29] is an extension to [28] by considering secure and reliable data collection. It improves the protocol's security by applying the secret sharing strategy. [30] proposes an efficient N-to-1 multipath routing protocol based on a minimum spanning tree and a learning mechanism.

[31] proposes an energy efficient collision aware multipath routing for WSN. It finds two collision-free paths to reduce the number of collisions among the sensor nodes in the network. [32] proposes a Low-Interference Energy-efficient Multipath Routing protocol (LIEMRO) for WSNs. This protocol aims at improving packet delivery ratio, lifetime, and latency by discovering multiple interference-minimized node-disjoint paths between source node and sink node.

[33] intends to improve the Direct Diffusion algorithm in order to allow multipath routing in a multimedia wireless sensor network which suffers from interferences. The base station selects paths which have disjoint node between them. [34] proposes a distributed, scalable and localized multipath search protocol to discover multiple node-disjoint paths between the sink node and source node, and a load-balancing algorithm to distribute the traffic over the multiple paths discovered.

All the previous multipath based routing approaches use static routing paths, which make it easy for the attackers to find the target sensor nodes for attacks [35].

2.2.3 Randomized Multipath Routing

The most related multipath routing is presented in [36], where the data packets are sent by each sensor towards the base station among three randomized dispersive multipath routes. The aim of their scheme is to avoid black-hole attacks by splitting the data packet into n shares. According to Shamir's algorithm, if k shares out of n are received by the base station, the original data packet can be correctly reconstructed. Therefore, the intention of this strategy is to have at least k shares received by the base station even when there is a black hole attack, thereby defeating the attack. [36] presents four distributed schemes for propagating shares: purely random propagation (PRP), directed random propagation (DRP), nonrepetitive random propagation (NRRP), and multicast tree-assisted random propagation (MTRP). PRP utilizes only one-hop neighborhood information and provides baseline performance. DRP utilizes two-hop neighborhood information to improve the propagation efficiency, leading to a smaller packet interception probability. The NRRP scheme achieves a similar effect in a different way. It records all traversed nodes to avoid traversing them again in the future. MTRP tries to propagate shares in the direction of the sink, making the delivery process more energy efficient.

In [12], the authors formulate the secret sharing-based multipath routing to increase the reliability and avoid black hole attack. The proposed algorithm is very close to [36]. Both algorithms build their solutions based on the location of the black hole, and Shamir's algorithm for secret sharing. The algorithm consists of three phases. In the first two phases, the scheme randomly delivers shares to all sensor nodes of the WSN and then attempts to transmit to the base station. The algorithm improves the security for single and multiple black hole attacks without reducing the network lifetime [37].

In [38], semi-randomized propagation for secure routing in WSN is introduced to defend against sink replication attack. The existing routing path can be used for transmission until the sink resets all the routes. Consequently, the base station establishes a new path

periodically.

According to [37], both data segmentation at the sender and the re-assembly at the base station will introduce overheads. Random selection of a path distributes the energy consumption among the explored paths. Accordingly, multipath routing increases and improves the reliability and the security by avoiding failure or compromised node. According to [31], flooding for route discovery with maximum transmission power wastes energy of sensor nodes. Also, the increase of the size of the packet costs extra energy in packet transmission. In addition, using an encryption/decryption scheme is expensive for WSNs due to their constraints. Unfortunately, security in WSN is always computationally expensive [39].

The previous randomized multipath routing protocols attempt to use encryption /decryption schemes and/or marking the path in the data packet by allowing signatures. We introduce randomized multipath routing for the selective forwarding attack and modification attack with less cost.

2.3 Our Approach

The target WSN is static, i.e., the location of each sensor is fixed. There is only one base station [1]. Each sensor node has a set of neighbouring sensor nodes with which it can communicate directly. Each communication link is bidirectional. The whole network is connected, i.e., for each sensor node, there is a routing path between this sensor node and the base station. When a sensor node generates a packet, this packet needs to be sent to the base station. No data aggregation is performed during data collection.

We investigate the problem of minimizing the failure rate of packet delivery in the presence of the modification attacks and the selective forwarding attacks in a static WSN with one

[1]Our approach is also applicable to a WSN with multiple base stations.

base station without using expensive encryption/decryption algorithms. Our objective is two folds. Firstly, we aim at minimizing the failure rate of packet delivery. Secondly, our approach makes it difficult for the attackers to attack the packets from a set of target sensor nodes.

Our approach consists of two major phases, namely, initialization phase and randomized multipath routing phase. During the initialization phase, our approach constructs a shortest path routing tree with the maximum lifetime proposed in [40], and assigns a unique ID to each sensor node. The ID of each sensor node will be used in randomized multipath routing. During the second phase, when a sensor node initiates a packet, our approach creates three copies and sends each copy to the base station via a different path. In order to make it difficult for malicious path analysis, our approach selects two routing paths at random. The 3rd copy of the packet travels to the base station along a static path.

2.3.1 Initialization Phase

During the initialization phase, our approach constructs a shortest path spanning tree T rooted at the base station with the maximum lifetime as proposed in [40], then assigns a unique ID to each sensor node in a distributed way. The ID of each sensor v_i, denoted by ID_i, is defined as follows.

- Let $v_{i_1}.v_{i_2}, \cdots, v_{i_k}$ be all the children of the base station in T sorted in anti-clock-wise order in the polar coordinate system with the base station as the pole.
- Assign a unique ID to each subtree rooted at a child of the base station in T. The ID of the subtree rooted at v_{i_j} is j.
- For each subtree rooted at a child of the base station in T, assign a unique rank to each sensor node in the subtree. The rank of a sensor node in a subtree is its rank in the depth-first traversal order of the subtree.

2. Randomized Multipath Routing for Secure Data Collection

- For each sensor node v_s, its ID, denoted by ID_s, is a tuple (x_s, y_s), where x_s is the ID of the subtree containing v_s rooted at a child of the base station in T, and y_s is the rank of v_s.

Next, we show how the ID of each sensor node is assigned in a distributed way. Our distributed naming algorithm consists of two phases. In the first phase, the base station creates a message for calculating the size of each subtree in T. This message will be sent to each sensor node along the tree T. When it reaches a leaf sensor node, the leaf sensor node will send an acknowledgment message to its parent in T. An acknowledgment message sent by a sensor node contains the size of the subtree rooted at the sensor node. When a sensor node receives the acknowledgment messages from all its children, it calculates the size of the subtree rooted at itself. In the second phase, the base station initiates a message for assigning a unique ID to each sensor node. This message carries the rank of the receiver of this message.

For each sensor node v_i, we introduce the following variables:

- ID: the ID of v_i with two fields x and y, where x is the ID of the subtree containing v_i rooted at a child of the base station in T, and y is the rank of v_i.
- $size$: the size of the subtree of T rooted at v_i.
- $depth$: the depth of v_i in the tree T. We assume that the depth of each sensor node is computed when our approach constructs a tree rooted at the base station with the maximum lifetime.

We use the following types of messages.

- COMPUTE-SUBTREE-SIZE($subtreeID$). This message is created by the base station, and sent to each sensor node in T. The parameter $subtreeID$ is the ID of the subtree rooted at a child of the base station.

- SUBTREE-SIZE($size$). $size$ is the size of the subtree of T rooted at a sensor node that sends this message. If a leaf node receives COMPUTE-SUBTREE-SIZE($subtreeID$), it will send SUBTREE-SIZE(1) to its parent. If a non-leaf sensor node receives a $SUBTREE-SIZE(size)$ from each child, it will calculate the size of the subtree of T rooted at itself, and send this message to its parent.

- COMPUTE-ID($rank$). This message is created by the base station and sent to each sensor node, where $rank$ is the rank of the receiver of this message. When a sensor node v_i receives this message, v_i will send a COMPUTE-ID($rank_j$) message to each child v_j, where $rank_j$ is the rank of v_j. The ranks of the children of v_i are computed as follows. Let $v_{j_1}, v_{j_2}, \cdots, v_{j_m}$ be all the children of v_i. The rank of v_{j_1} is equal to $rank_i + 1$, where $rank_i$ is the rank of v_i. The rank of $v_{j_s}(s = 2, \cdots, m)$ is equal to $rank_{j_{s-1}} + size_{j_{s-1}}$, where $rank_{j_{s-1}}$ is the rank of $v_{j_{s-1}}$, and $size_{j_{s-1}}$ is the size of the subtree rooted at $v_{j_{s-1}}$.

The details of the initialization phase are shown in Algorithm 1.

========== **Algorithm 1: Initialization phase** ==========

========== For the base station:==========

Let L be a list of all the children of the base station in T sorted in anti-clock-wise order in the polar coordinate system with the base station as the pole;

$subtreeID = 1$;

for each child v_i in L **do**

  Send COMPUTE-SUBTREE-SIZE(subtreeID) to v_i;

  subtreeID=subtreeID+1;

end for

$i = 1$;

while $i \leq |L|$ **do**

  if SUBTREE-SIZE($size_i$) is received from a child v_i **then**

Send COMPUTE-ID(1) to v_i;

$i = i + 1$;

end if

end while

========== For each sensor node v_i:==========

$v_i.size = 0$;

Receive COMPUTE-SUBTREE-SIZE(subtreeID) from the parent;

$v_i.x = subtreeID$;

if v_i is not a leaf node **then**

for each child v_j of v_i **do**

Send COMPUTE-SUBTREE-SIZE(subtreeID)

to v_j;

end for

for each child v_j of v_i **do**

Receive SUBTREE-SIZE($v_j.size$) from v_j;

$v_i.size = v_i.size + v_j.size$;

end for

else

$v_i.size = 1$;

Send SUBTREE-SIZE($v_i.size$) to the parent;

end if

Receive COMPUTE-ID($rank$);

$v_i.y = rank$;

$rank = rank + 1$;

if v_i is not a leaf node **then**

for each child v_j of v_i **do**

Send COMPUTE-ID($rank$) to v_j;

$rank = rank + v_j.size;$

end for

end if

========== **End of Algorithm 1**==========

Figure 2.1 shows the spanning tree T and the IDs of all the sensor nodes after the initialization phase is completed.

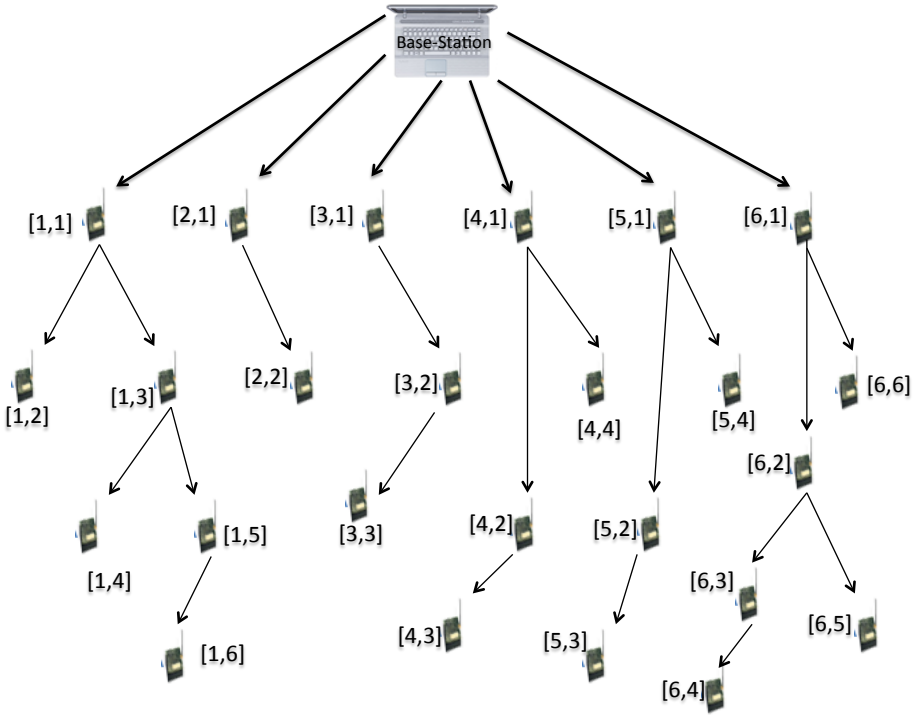

Figure 2.1: An Example Of The Initialization Phase

2.3.2 Randomized Multipath Routing Phase

After constructing the spanning tree T and assigning a unique ID to each sensor node, each sensor node can start sending its packets to the base station.

When a sensor node generates a packet to be delivered to the base station, it creates three

copies of the packet and sends the three copies along three paths to the base station. In order to make it difficult for the malicious nodes to attack the packets from certain target sensor nodes, our approach constructs two paths at random. Specifically, when a sensor node v_i creates a packet, it generates two natural numbers X and Y between 1 and the maximum ID of the subtrees rooted at the base station's children by using a random number generator. The first copy is sent to the base station along the path from v_i to the base station in T. The two paths for the second copy and the third copy are selected according to the Selection Scheme.

Selection Scheme

- Assume that the second copy is currently at a sensor node v_p. If v_p is a child of the base station, v_p will send the second copy to the base station. Otherwise, v_p will check if the second copy has visited X different subtrees each of which is rooted at a child of the base station. If it is the case, v_p will send the second copy to its parent in T. Otherwise, it will do the following. Let v_t be the child of the base station that is an ancestor of v_p in T. v_p tries to find a neighbouring sensor node v_j as a candidate receiver of the second copy satisfying the following constraints.

  1. v_j is in a different subtree rooted at a child v_s of the base station.
  2. The ID of the subtree rooted at v_s is smaller than that of the subtree rooted at v_t.

  If such a candidate receiver v_j is not found, v_p will send the second copy to its parent in T. Otherwise, v_p will check if there is only one candidate receiver satisfying the above constraints. If it is the case, v_p will send the second copy to the candidate receiver. Otherwise, among all the candidate receivers, v_p will find the sensor node v_r with the smallest depth in T, and send the second copy to v_r.

- Assume that the third copy is currently at a sensor node v_p. If v_p is a child of the base station, v_p will send the second copy to the base station. Otherwise, v_p will

check if the third copy has visited Y different subtrees each of which is rooted at a child of the base station. If it is the case, v_p will send the third copy to its parent in T. Otherwise, v_p will do the following. Let v_t be the child of the base station that is an ancestor of v_p in T. v_p tries to find a neighbouring sensor node v_j as a candidate receiver of the third copy satisfying the following constraints.

1. v_j is in a different subtree rooted at a child v_s of the base station.
2. The ID of the subtree rooted at v_s is larger than that of the subtree rooted at v_t.

If such a candidate receiver v_j is not found, v_p will send the third copy to its parent. Otherwise, it will check if there is only one candidate receiver satisfying the above constraints. If it is the case, v_p will send the third copy to the candidate receiver. Otherwise, among all the candidate receivers, v_p will find the sensor node v_r with the smallest depth in T, and send the second copy to v_r.

We introduce the following message for randomized multipath routing.

- PACKET-DELIVERY(*creator*, *copyid*, *packetid*, *subtreestogo*, *packetdata*). This message is used to send any one of the three copies of a packet to the base station, where *creator* is the creator of the packet, *copyid* has one of the three values, 1, 2 and 3, indicating one of the three copies, *packetid* is a unique ID of the packet, *subtreestogo* denotes the number of subtrees rooted at the base station's children that this copy needs to visit, and *packetdata* is the packet itself.

The details of randomized multipath routing are described in Algorithm 2. The base station accepts a packet only if it receives at least two identical copies.

====== **Algorithm 2: Randomized multipath routing phase**=====

========== For the base station:==========

if a copy of a packet is received for the first time **then**
  set a timer for this copy;
  if the timer expires **then**
    if at least two identical copies have been received **then**
      Accept any one of the identical copies;
    else
      Reject the packet;
    end if
  end if
end if

========== For each sensor node v_i:==========

/* Each packet has three copies *copy*1 ,*copy*2, and *copy*3 sent to the base station via three paths */

if v_i generates a packet **then**
  Create three copies, *copy*1, *copy*2 and *copy*3 of the
  packet;
  Generate two natural numbers X and Y between 1 and
  the maximum ID of the subtrees rooted at the base
  station's children by using a random number
  generators;
  Generate a unique ID for the packet;
  packetid = the ID of the packet;
  creator = the ID of v_i;
  packetdata = data of this packet;
  for *copyid* $= 1, 2, 3$ **do**

switch *copyid* **do**

case 1:

$subtreestogo = 0$;

case 2:

$subtreestogo = X$;

case 3:

$subtreestogo = Y$;

Find the receiver of this copy;

Send PACKET-DELIVERY(*creator*, *copyid*,

packetid, *subtreestogo*, *packetdata*) to the

receiver;

end for

end if

if PACKET-DELIVERY(*creator*, *copyid*, *packetid*, *subtreestogo*, *packetdata*) is received **then**

switch *copyid* **do**

case 1:

Send PACKET-DELIVERY(*creator*, *copyid*,

packetid, *subtreestogo*, *packetdata*) to the

parent in T;

case 2, 3:

Find the receiver as discussed before;

if the receiver is in a different subtree rooted at a child of the base station **then**

$subtreestogo = subtreestogo - 1$;

end if

Send PACKET-DELIVERY(*creator*, *copyid*,

packetid, *subtreestogo*, *packetdata*) to the

receiver;

end if

========== **End of Algorithm 2**==========

2.3.3 A Comprehensive Example

Next, we use an example to illustrate how our approach works. Consider a WSN with a spanning tree shown in Figure.2.2, where a dashed line denotes a communication link not in the tree. Assume that the sensor node $[4, 3]$ generates a packet to be sent to the base station. Our approach will create three copies, *copy*1, *copy*2 and *copy*3. Assume that the two natural numbers X and Y generated by a random number generator are 2 and 1, respectively. *copy*1 will be sent to the base station via the deterministic path $[4, 3] \rightarrow [4, 2] \rightarrow [4, 1] \rightarrow$ base station. As to *copy*2, the sensor node $[4, 3]$ will select the sensor node $[3, 3]$ as the receiver of *copy*2 as it is the only candidate receiver. Next, the sensor node $[3, 3]$ will select the sensor node $[1, 5]$ as the receiver of *copy*2. So far, *copy*2 has traversed X subtrees rooted at the base station's children. Therefore, *copy*2 will be sent to the base station along the path $[1, 5] \rightarrow [1, 3] \rightarrow [1, 1] \rightarrow$ base station in the current subtree rooted at the sensor node $[1, 1]$. As to *copy*3, the sensor node $[4, 3]$ has two candidate receivers, $[5, 3]$ and $[5, 2]$. Since $[5, 2]$ has a smaller depth in the tree T, $[5, 2]$ will be selected as the receiver of *copy*3. After *copy*3 is sent to $[5, 2]$, *copy*3 has traversed Y subtree. Therefore, *copy*3 will be sent to the base station along the path $[5, 2] \rightarrow [5, 1] \rightarrow$ base station. Notice that if any one of the three copies is compromised or dropped, the base station can still receive the correct packet.

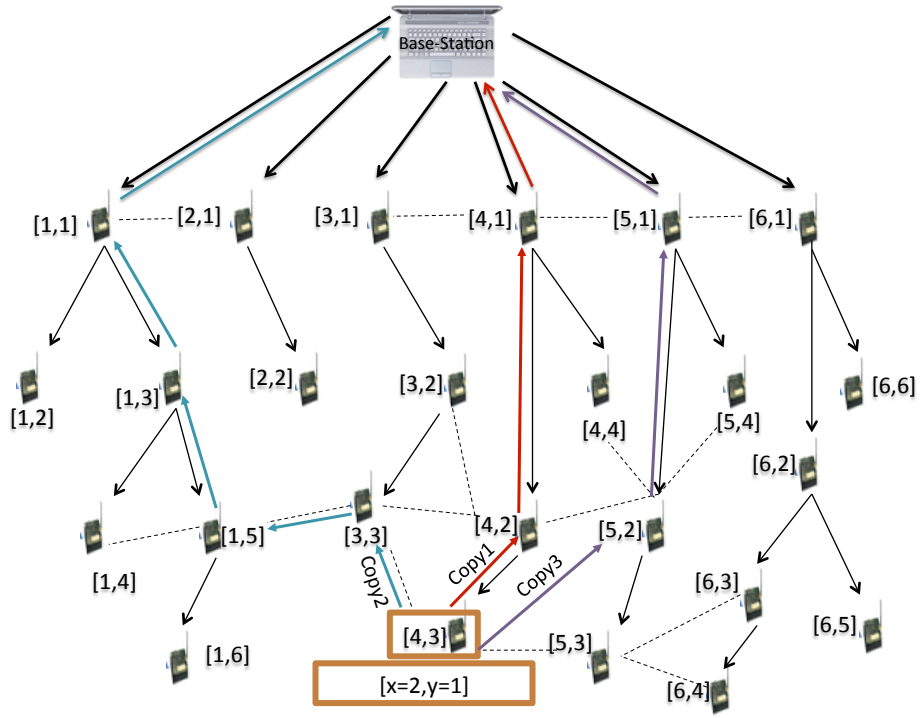

Figure 2.2: An Example Of Randomized Multipath Routing

2.4 SIMULATION RESULTS

2.4.1 Setup

In order to evaluate our approach, we compare it with the state-of-the-art approach Multicast-Tree Random Propagation (MTRP) proposed in [36] for the following two reasons. Firstly, both approaches use randomized multipath routing. Secondly, MTRP is the most recent work on randomized multipath routing for secure data collection and appears in a prestigious journal. However, the main objectives are different. Our approach (SMRP) focuses on minimizing the failure rate in presence of modification and selective forwarding attacks while their approach focuses on delivering k shares out of n in the presence of a black hole attack. MTRP splits data into multiple shares. In the simulations, we choose 4 shares for MTRP.

Three performance metrics are used. The first metric is the failure rate which is defined as the percentage of packets incorrectly received by the base station. A packet is considered incorrect, if at least two copies of the packet have been received and forwarded by a malicious node. The second metric is the network lifetime which is the time when the first sensor node depletes its energy. The third metric is the delay which represented by the number of intermediate node between a source node and the base station.

We generate 20 instances of WSNs using NS- 2.35 simulator. The number of sensor nodes of each instance ranges from 100 to 1050, with an increment of 50 sensor nodes. Each of the 20 instances, generated 20 different times on random distribution. When generating an instance, we ensure the network's tree is connected. The transmission range is fixed to 50 meters. In each instance, sensor nodes are randomly deployed in a square area with a size of 400 x 300 square meters. The energy consumption on sending one bit of data is $TX = 31.28 * 10^{-3} \mu J$ while the energy consumption on receiving one bit of data is $RX = 35.28 * 10^{-3} \mu J$ while the initial energy for every sensor node is $E_i = 500J$.

The hardware platform is Intel Core i5-2500 with a clock speed of 3.30 Ghz, a memory size of 8 GB and a cache size of 6144 MB.

After constructing each instance, we randomly select a set of malicious sensor nodes among all the sensor nodes that are not the neighbours of the base station. A number of percentages of malicious sensor nodes are used, ranging from 0.1% to 1.9%, with an increment of 0.2%. Each sensor node generates one packet to the base station per unit time. For each instance, each sensor node generates $x * 3$ packets during the simulation, where x is the number of sensor nodes in the instance.

2.4.2 Simulation Results

This section show the simulation results and analysis for each of the three performance matrices.

Failure Rate Comparison

Failure rate is the primary performance metric for measuring the efficiency of our approach to the secure data collection problem. The lower the failure rate, the better the approach.

Figures 2.3 and 2.4 show the failure rates for different percentages of malicious nodes. In each figure, the horizontal axis denotes the number of sensor nodes in each instance, and the vertical axis denotes the failure rate for each instance and each approach.

According to the simulation results, when the percentage of malicious sensor nodes is low, the failure rate is very low in our proposed approach (SMRP) while the failure rate in the other approach (MTRP) varies and is usually higher than SMRP. As shown in Figure 2.3, the maximum failure rate occurs in SMRP is about 6% in the instance with 400 sensor nodes while it reaches almost 16% in MTRP when the instance has 500 sensor nodes. However, in Figure 2.4, SMRP reaches over 10% in its maximum failure rate when the instance has 900 sensor nodes and MTRP reaches over 18% when the instance has 1000 sensor nodes.

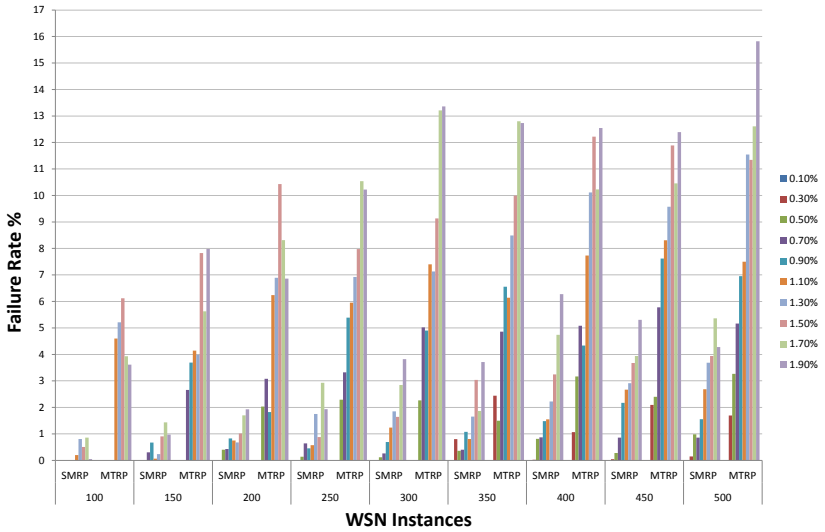

Figure 2.3: Failure Rate Comparison For All Instances From 100 To 500

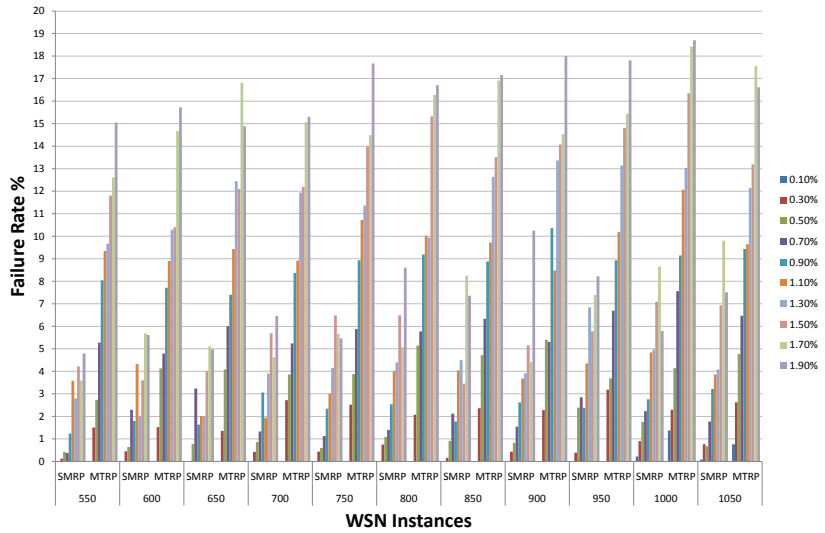

Figure 2.4: Failure Rate Comparison For All Instances From 550 To 1050

Figure 2.5 shows the improvement of SMRP over MTRP in terms of failure rate for the instances with 100, 500, and 1000 sensor nodes. From the simulation results, SMRP performs better than MTRP in all the scenarios. Especially, the maximum improvement of almost 100% is achieved for the instance with 100 nodes.

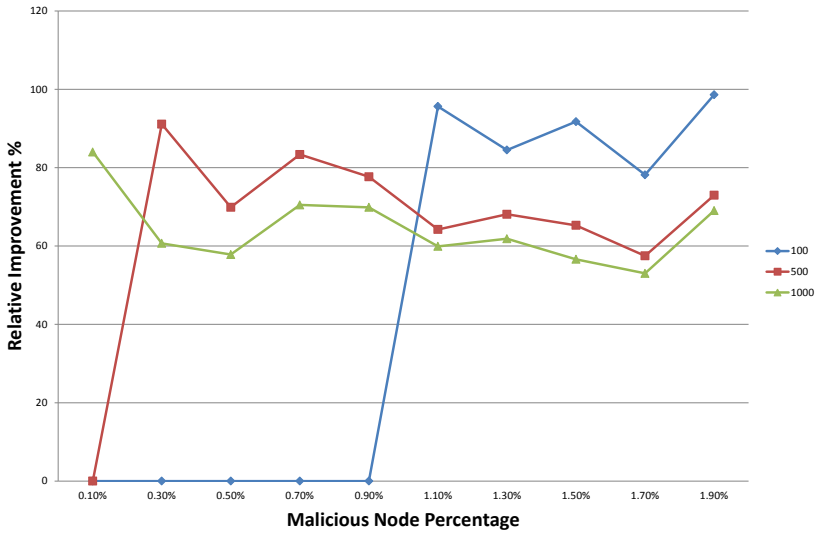

Figure 2.5: Failure Rate Improvements

In general, in terms of the failure rate, the minimum improvement, the maximum improvement and the average improvement are 35.224%, 98.603%, and 72.896%, respectively.

From the simulation results for the failure rate, we have the following key observations:

1. In general, as expected, the failure rate increases as the percentage of malicious sensor nodes increases.

2. For SMRP, given a fixed percentage of malicious sensor nodes, the failure rate increases as the size of an instance increases. The reason is as follows. When the number of sensor nodes increases, the height of the routing tree also increases. If a

malicious sensor node is close to the base station, the packets from all its descendants will be attacked, resulting in an increased failure rate. For MTRP, the same can be noticed.

3. The failure rate is not very consistent across all the instances for both approaches. The inconsistency exists due to the following major reasons. Firstly, the malicious sensor nodes are selected at random. Clearly, the location of a malicious sensor node is very important. A malicious sensor node close to the base station may have a much higher impact on the failure rate than a malicious sensor node far away from the base station. Secondly, each WSN instance is generated at random. The structure of the spanning tree T has a significant impact on the failure rate. A malicious sensor node with more descendants in the tree T has a higher impact on the failure rate than a malicious sensor node with fewer descendants. However, we can notice that SMRP and MTRP have qualitatively similar results and vary in very rare cases due to the above reasons.

4. A larger random number in MTRP causes more hops and requires more random relays, which increase the number of intermediate nodes and accordingly results in a higher possibility for a packet to be attacked.

5. The proposed naming approach in SMRP plays an important role in limiting the failure rate because of the range it provides to the generated numbers x and y.

Network Lifetime Comparison

Network lifetime is another important performance metric. A good approach to the secure data collection problem should maximize the network lifetime. The network lifetime measured based on the total energy consumption on (receiving, and sending one bit of data * the total number of bits).

Figures 2.6 shows the simulation results for network lifetime. In Figure 2.6, the horizontal

axis denotes the number of sensor nodes of each instance, and the vertical axis denotes the total lifetime in time unit. From the simulation results, SMRP results in much less energy than MTRP. Figure 2.7 shows the network lifetime improvements , where the horizontal axis denotes the percentage of malicious nodes, and the vertical axis denotes the network lifetime improvements in percentage of SMRP over MTRP. The simulation results show that SMRP achieves improvements in all the scenarios.

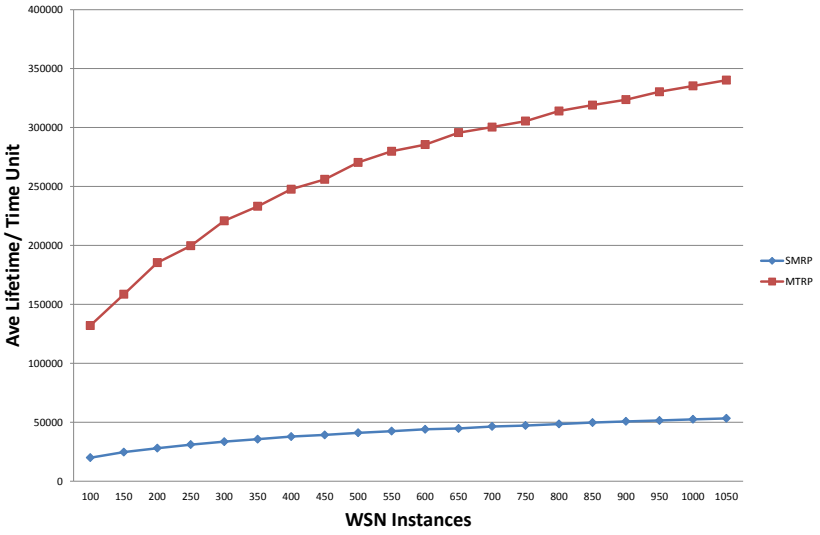

Figure 2.6: Average Network Lifetimes

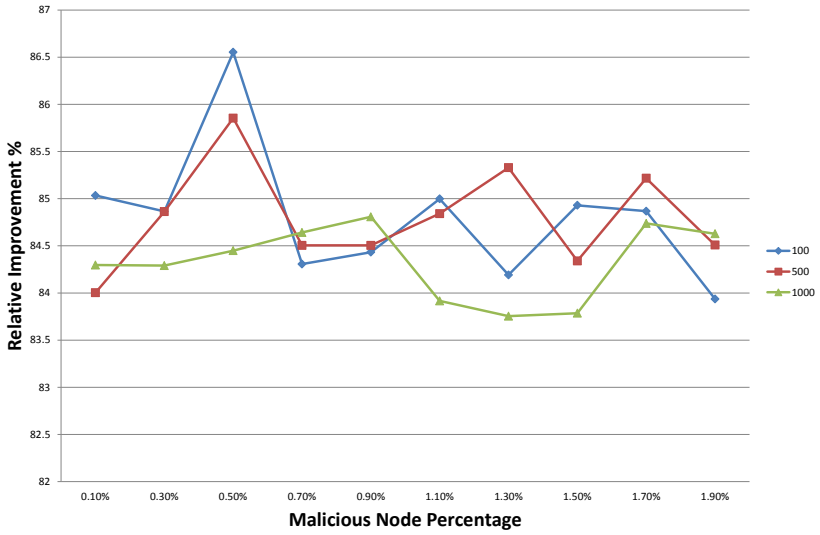

Figure 2.7: Network Lifetime Improvements

Overall, the minimum improvement, the maximum improvement, and the average improvement in terms of network lifetime are 82.68%, 86.55%, and 84.579%, respectively. From the simulation results, we make the following key observations.

1. The major factors which affects the network lifetime are the packet size, the number of packets, and the number of hops.

2. MTRP produces more packets than SMRP as it splits sensed data into multiple shares. As a result, it consume more energy for sensor nodes to relay those packets.

3. On the one hand, MTRP splits data into shares, which reduces the data size of each packet. On the other hand, each packet in MTRP has other fields,such as the IDs of all the neighbours of the receiver/forwarder, which may increase the overall packet size.

4. In MTRP, a routing path may not be the shortest one. In contrast, SMRP always

finds a shortest path for each sensor. The energy consumption of sensor nodes in increases as the number of hops increases. Therefore, SMRP results in less energy consumption of sensor nodes.

Latency Comparison

In many applications such as battle field monitoring, real time delivery of data across the network is essential. Many factors, including the number of hops, network congestion, and sensor node failures, may affect the latency. We use the hop count to measure the latency as network congestion and sensor node failures occur rarely. For each instance, we compute the average latency of all the sensor nodes.

Figure 2.8 shows the average latency for SMRP and MTRP, where the horizontal axis denotes the number of sensor nodes of each instance, and the vertical axis denotes the average number of hops. Figure 2.9 shows the improvements of SMRP over MTRP in terms of average latency, where the horizontal axis denotes the percentage of malicious nodes, and the vertical axis denotes the improvements in percentage of SMRP over MTRP for the instances with 100, 500, and 1000 sensor nodes.

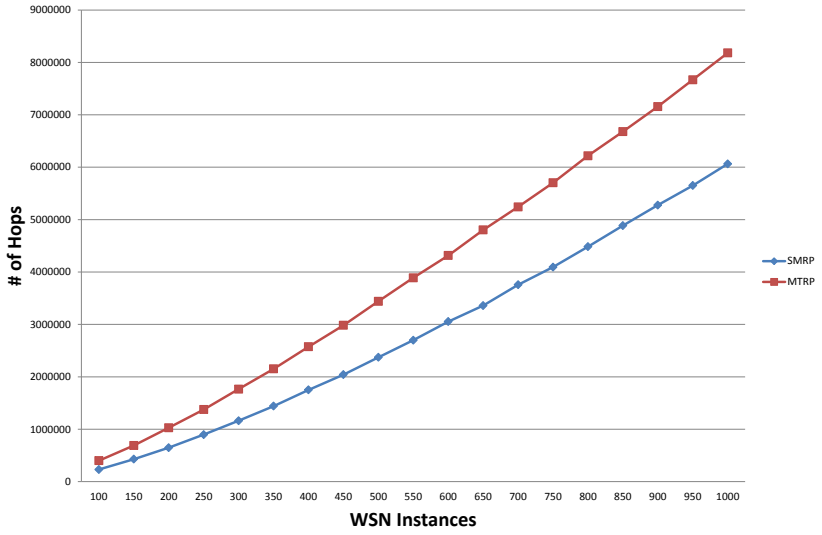

Figure 2.8: Average Latency Comparison

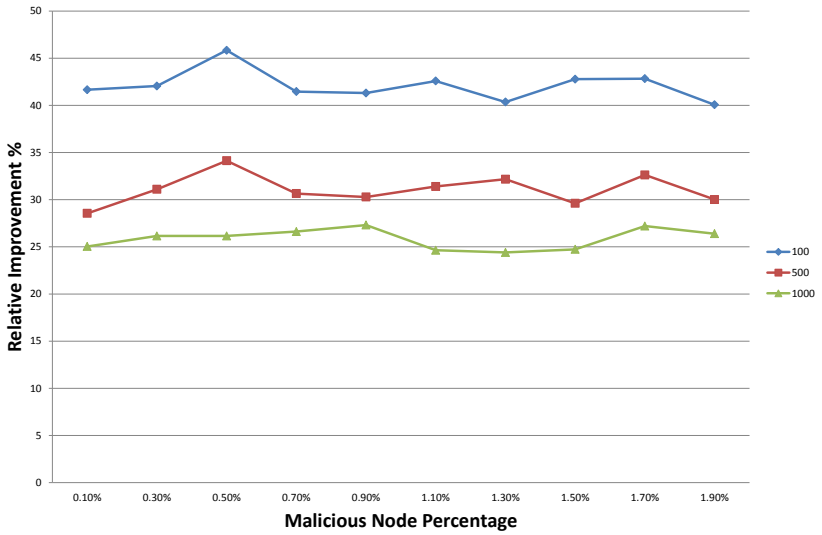

Figure 2.9: Latency improvements

From the simulation results, we notice that the latency increases as the network size increases and that SMRP has a lower average latency over all instances. Furthermore, the performance gap between SMRP and MTRP increases as the network size increases. For all the instances, the average latency in SMRP is lower than MTRP. The minimum improvement, the maximum improvement, and the average improvement in terms of latency are 22.94%, 45.839%, and 30.903%.

From the simulation results for latency, we have the following key observations:

1. As expected, the average latency increases in both SMRP and MTRP whenever the network size increase. However, SMRP achieves a lower average latency than MTRP. The key reason is that SMRP always finds a shortest path while MTRP selects a random path when a sensor nodes sends its sensed data to the base station.

2.5 Chapter's Summary

We investigate the problem of minimizing the failure rate of packet delivery in the presence of the modification attacks and the selective forwarding attacks in a static WSN with one base station, and propose a novel heuristic approach. Our approach consists of two phases, namely the distributed naming phase and the randomized multipath routing phase. The distributed naming phase constructs a shortest path routing tree and assigns a unique ID to each sensor node. the randomized multipath routing phase, each sensor nodes makes three identical copies of its data packet, and sends the three copies along three different paths, one fixed shortest path and two random shortest paths.

We have compared our approach with a state-of-the-art approach using three performance metrics, namely the failure rate, the network lifetime and the average latency. Simulation results show our approach significantly improves the state-of-the-art approach in terms of all the three performance metrics.

2. Randomized Multipath Routing for Secure Data Collection

In this approach, we decide to have at least three copies of the same packet for the following reasons. First, it allows the base station (BS) to compare between at least two identical packets to determine the correct packet. This cannot be done by sending two copies as the base station does not have a reference to compare against. On the other hand, although having more than 3 copies of the same packet may reduce the failure rate of packet delivery. Excessive network traffic due to more packets shortens the network lifetime and possibly causes partitioning. Using multipath routing with three-copy strategy makes a good trade-off between the failure rate of packet delivery and network lifetime. Compared with a single copy approach, the three-copy approach may increase information leakage. However, it may significantly reduce the failure rate of packet delivery as any attack on any one of the three copies does not affect the correctness of data received by the base station.

Chapter 3

Constructing A Shortest Path Overhearing Tree With Maximum Lifetime

Secure data collection is an important problem in wireless sensor networks. Different approaches have been proposed. One of them is overhearing. In this chapter, we investigate the problem of constructing a shortest path overhearing tree with maximum lifetime. We propose three approaches. The first one is a polynomial-time heuristic. The second one uses ILP (Integer Linear Programming) to iteratively find a monitoring node and a parent for each sensor node. The last one optimally solves the problem by using MINLP (Mixed-Integer Non-Linear Programming). Furthermore, we evaluate the three approaches via extensive simulations.

3.1 Introduction

A WSN (Wireless Sensor Network) consists of a large number of autonomous sensors nodes and a single or multiple base stations. Each sensor node delivers the data sensed from the physical environment to its designated base station. Typically, sensor nodes are battery powered. Most of the energy of a sensor node is consumed by communication. In order to save energy, the transmit power of each sensor node is kept low, leading to a short transmission range. Thus, data collection is performed in a multi-hop way.

In WSNs, various attacks may exist. Among them are selective forwarding attacks and modification attacks. In the selective forwarding attacks, a malicious sensor node may deliberately drop some packets received from other sensor nodes, resulting in packet loss. In the modification attacks, a malicious sensor node may modify some packets received from other sensor nodes and forward the incorrect packets to the base station. In order to ensure that the data sensed by each sensor are delivered to the base station correctly, the protocols for secure routing are required.

The secure and reliable data collection problems in WSNs have been extensively investigated. Many approaches are based on the overhearing technique [23]. When a sensor node receives a packet and forwards it to another sensor node, a third sensor node will overhear the packet reception and transmission. Therefore, the overhearing technique can be used to detect the modification attacks and the selective forwarding attacks.

We investigate the problem of constructing a routing and overhearing topology with the maximum network lifetime in a WSN with a single base station. Specifically, we construct a shortest path tree for routing and select a monitoring sensor node for each sensor node such that the network lifetime is maximized. For each sensor node v_i, its monitoring sensor node overhears the reception and transmission of each packet that the parent of v_i receives from v_i. The network lifetime is defined as the time when the first sensor node depletes its energy. Even without considering overhearing, the problem of constructing a shortest

path tree with the maximum lifetime is NP-Complete [41].

We make the following major contributions.

- We propose a polynomial-time heuristic approach, an ILP-based approach, and a MINLP-based approach to the problem of constructing a shortest path overhearing tree with the maximum network lifetime for both homogeneous WSNs and heterogeneous WSNs. To the best of our knowledge, our work is the first attempt to construct such an overhearing topology with the maximum lifetime for security purposes.
- We have implemented our approaches using MIDACO solver and MATLAB Intlinprog, and performed extensive simulations on 150 network instances with three different distributions, namely, uniform, grid, and random distributions. For homogeneous networks, the simulation results show that the average lifetime of all the network instances achieved by the heuristic approach is 85.69% of that achieved by the ILP-based approach and 81.05% of that obtained by the MINLP-based approach, and the performance of the ILP-based approach is almost equivalent to that of the MINLP-based approach. For heterogeneous networks, the simulation results show that the average lifetime of all the network instances achieved by the heuristic approach is 87.371% of that achieved by the ILP-based approach and 74.9% of that obtained by the MINLP-based approach, and the performance of the ILP-based approach is almost equivalent to that of the MINLP-based approach.
- One limitation with our shortest path overhearing tree is that if a monitor colludes with the parent of the monitored sensor node, the attacks by the parent of the monitored sensor node may not be detected. We overcome this by selecting a different monitor for every sensor node in every time interval.

The rest of the chapter is organized as follows. Section 3.2 gives a brief survey of the related work. Section 3.3 provides the network model and definitions. Section 3.4 describes our

three approaches for constructing a shortest path overhearing tree with maximum lifetime in homogeneous WSNs and their extensions to heterogeneous WSNs in Section 3.5 while 3.6 provide a discussion on a security thread. Section 3.7 shows the simulation results and analyses. Lastly, section 3.8 concludes this chapter.

3.2 Related Work

The overhearing technique has been widely used to improve the security and the reliability of data collection in WSNs. The problem of constructing a routing tree with the maximum network lifetime has also been extensively investigated. Next, we give a survey of the major work related to secure and reliable data collection by using overhearing and the lifetime-aware routing tree construction problem in WSNs.

3.2.1 Overhearing-Based Secure and Reliable Data Collection

[23] uses an overhearing technique to detect modification attacks in WSNs. By the overhearing technique, a committee structure is constructed for each sensor node. The committee structure includes several committee sets, and each committee set is designed for a specific communication link. Due to the microwave nature of the wireless channel, neighbouring sensor nodes within a sender's radio range can overhear the packet the sender is transmitting. Therefore, each packet can be examined by the sensor nodes of the committee set during forwarding. If a packet is modified by a malicious node, the committee will detect the anomaly. [42] provides several drawbacks of the security mechanism used by [23]. Firstly, there is no mechanism implemented for neighbour nodes authentication within the construction phase. Hence, the malicious node may penetrate into the network to send fake information about their neighbours and contribute in voting. Secondly, in the last phase, the proposed mechanism does not isolate the malicious nodes from the network.

[43] proposes an efficient overhearing-based reliable transfer protocol for WSNs. The monitoring sensor node overhears a packet being transmitted by the monitored sensor node to determine whether the monitored node is malicious or not. To make such a decision, a reputation value is calculated for each sensor node in the designated area. The sensor node with a reputation value below the threshold is considered as a malicious sensor node.

[44] proposes an efficient, reliable transfer protocol for WSNs by introducing implicit and selective acknowledgment. The selective acknowledgment mechanism is executed by comparing the current path reliability and the base reliability. Overhearing is also used in the implicit acknowledgment mechanism.

Several approaches [45–48] use the overhearing technique to avoid redundant information to be transmitted towards the base station, improving the energy efficiency.

[45] proposes an energy-efficient data transmission reduction approach for periodical data gathering in WSNs by using the overhearing technique. In this approach, each sensor node in a WSN autonomously determines whether its own reading is redundant or not by using the overheard packets transmitted by its neighbors. If the reading is determined as redundant, the sensor node stops transmitting it. [47] extends this approach by proposing an overhearing-based data aggregation method using spatial and temporal interpolations.

[46] presents OBMAC, an enhancement for MAC protocol based on the overhearing technique in WSNs. The objective of the proposed protocol is to reduce the number of redundant packets using the overhearing technique. In OBMAC, every sensor node verifies each overheard packet and compares it to its own in order to avoid transmitting the same information to the base station. The notion of influential range is used to improve the efficiency of OBMAC.

[48] presents a method for evaluating the approach proposed in [45] by using a practical model of lossy links. The evaluation results show that the proposed approach suppresses

data transmissions and reduces total energy consumption even in a lossy environment.

[49] investigates the problem of improving the data persistence. It introduces a distributed scheme based on LT (Luby Transform)-codes and an overhearing technique. In the proposed scheme, each sensor node uses overhearing to check if a packet has been transmitted by one of its neighbors. When a sensor node needs to transmit a packet, it randomly chooses one of its neighbors that does not transmit the packet as the receiver. Each sensor node computes a key parameter of LT codes by using some properties of the packet transmission mechanism, and then stores the data accordingly. After the process of storage is finished, a collector will recover all the data by visiting a small subset of sensor nodes.

3.2.2 Lifetime-Aware Routing Trees

A routing tree of a WSN with a single base station is a spanning tree rooted at the base station. Each sensor node sends its own data and the data received from its children to its parent. Since each sensor node is typically battery-powered, it is important to construct a routing tree such that the network lifetime is maximized. Many approaches to the lifetime-aware routing tree construction problem have been proposed.

[50] shows that the problem of constructing a spanning tree with the maximum lifetime is NP-complete and proposes a polynomial-time approximation algorithm. The approximation algorithm starts with an arbitrary tree and iteratively reduces the loads of bottleneck nodes. [51] studies an on-line data gathering problem, and proves that the problem is NP-complete. It presents a generic cost model of energy consumption for data gathering queries and several heuristics.

[52] and [53] take into account the remaining energy and the load of each node, and propose top-level load balancing algorithms with dynamic modifications. [54] constructs a multi-tree topology to allow more choices for the next hop when routing messages. [55]

proposes a distributed probabilistic load-balancing converge cast tree algorithm to address the heterogeneity issues in terms of nodal traffic burden and residual energy by dynamically forming converge cast routing trees.

[56] proposes a new weighted path cost function improved from the shortest path tree approach. In this approach, links are assigned weights according to their path lengths to the root, and those close to the root have larger weights. By balancing loads according to the link weights, this approach increases the network lifetime compared with those randomly constructed shortest path trees.

[41] investigates the problem of finding a shortest path aggregation tree with the maximum lifetime in a WSN. The proposed algorithm first builds a fat tree which contains all the shortest path trees. Then, it converts the problem into a sequence of semi-matching problems each considering two adjacent levels of the fat tree, and solves each semi-matching problem by using the min-cost max-flow approach in polynomial time. [41] also proposes a distributed algorithm for constructing a maximum lifetime shortest path aggregation tree, and proves that if no data aggregation is performed, it is NP-complete to construct a shortest path tree with the maximum lifetime.

[57] investigates the lifetime-aware data collection problem without data aggregation, and proposes an approximation algorithm for constructing a routing tree with the maximum network lifetime. The approximation algorithm iteratively transfers some of the descendants of the node with the largest weight to a node with a smaller weight, and stops when no more descendants of a bottleneck node can be transferred.

[58] studies the load balance problem in a grid topology. It focuses on the energy consumptions of the nodes which can communicate with the base station directly. Firstly, the algorithm selects the most lightly loaded and most confined branches for growth. Secondly, it selects the heaviest nodes with the maximum growth space. After establishing a loosely balanced tree, the algorithm re-balances the tree by moving nodes from the heavi-

est loaded branches to more lightly loaded neighbouring branches. The simulation results show that the routing trees constructed by their algorithm are more balanced than the shortest path tree constructed by Dijkstra's algorithm.

[59] investigates the problem of network lifetime maximization of WSNs in the context of data collection trees. It proposes an efficient algorithm, called Randomized Switching for Maximizing Lifetime (RaSMaLai) that aims at maximizing the lifetime of WSNs through load balancing with a low time complexity, and a distributed version of the algorithm.

[60] investigates the problem of lifetime and latency-aware data collection in WSNs with one base station. It proposes a new routing structure, namely k-tree, and a distributed algorithm for constructing a lifetime-aware k-tree. A unique feature of the k-tree is that it provides the maximum latency guarantee for data collection.

3.3 Network Model and Definitions

The target WSN consists of a set $V = \{v_1, v_2, \cdots .v_n\}$ of n static sensors. There is only one base station. Each sensor generates one packet of data per unit time and sends the packet to the base station without performing any data aggregation. All the sensor nodes are identical with the same transmission range and the same initial energy level in homogeneous network and different with various initial energy levels in heterogeneous network. The base station is aware of the location of each sensor node. Since this is a construction phase, it can be done offline. The algorithm is performed in each sensor node and in the base station as well.

We use an undirected graph, named as connectivity graph, to represents the connectivity between sensor nodes in the WSN. The connectivity graph G is defined as follows: $G = (V \cup \{BS\}, E)$, where, BS is the base station, and $E = \{(v_i, v_j)$: $v_i, v_j \in V \cup \{BS\}$ and v_i and v_j can communicate with each other directly $\}$. A communication link between

two sensor nodes indicates that the two sensor nodes can communicate with each other. We assume that the connectivity graph is connected. The base station can collect the connectivity graph from all the sensor nodes.

Some sensor nodes may be compromised. A compromised sensor node is called a malicious sensor node. A malicious sensor node may drop, or modify the packets it receives from other sensor nodes.

Given the connectivity graph $G = (V, E)$ of a target WSN, the problem we investigate is to construct a shortest path routing tree rooted at the base station and assign a monitoring sensor node to each sensor node such that the network lifetime is maximized. Hereinafter, a monitoring sensor node is called a monitor. A monitor v_i of a sensor node v_k is used to detect if the data sent by v_k to its parent v_j will be forwarded correctly by v_j to its parent in the shortest path tree. Therefore, the monitor v_i needs to overhear the transmission of each packet v_k sends to its parent v_j and the forwarding of each packet v_j receives from v_k. If v_j drops or modifies any packet received from v_k, v_i will detect it and report it to the base station.

For each sensor node, its parent and monitor must satisfy the following requirements:

1. There is a communication link between the parent and the monitor.
2. There is a communication link between the monitor and the sensor node.
3. There is a communication link between the parent and the sensor node.
4. The parent and the monitor have the same depth in the routing tree.

When constructing a shortest path tree, we mainly consider the energy consumption of data receptions and transmissions for each sensor node. The energy consumption of listening and communication session setup is ignored. In heterogeneous WSNs, for each sensor node v_i, α_i is the energy consumed to receive one packet, and β_i is the energy consumed

to transmit one packet. In homogeneous WSNs, for each sensor node v_i, α is the energy consumed to receive one packet, and β is the energy consumed to transmit one packet. Given a set S, $|S|$ denotes the size of S. In heterogeneous WSNs, each sensor v_i has its own initial energy E_i. In homogeneous WSNs, all the sensor nodes have the same initial energy.

3.4 Constructing A Shortest Path Overhearing Tree With Maximum Lifetime In Homogeneous WSNs

3.4.1 Heuristic Approach

Our heuristic approach consists of three phases, namely, partitioning phase, initial monitor and parent selection phase, and energy balancing phase.

In the partitioning phase as shown in Figure 3.1, our heuristic approach partitions all the sensor nodes into m disjoint groups $C_i(i = 1, 2, \cdots, m)$ such that the shortest path length between each sensor node in the group C_i to the base station in the connectivity graph is equal to i. In the initial monitor and parent selection phase as shown in Figure 3.2, for each group $C_i(i = m, m-1, \cdots, 2)$, our heuristic approach assigns a parent and a monitor from the group C_{i-1} to each sensor node in the group C_i, aiming at minimizing the maximum energy consumption of all the individual sensor nodes in the group C_{i-1}. In the energy balancing phase as shown in Figure 3.3, our heuristic approach performs the monitor and parent adjustment for each group $G_i(i = m, m-1, \cdots, 2)$ such that all the sensor nodes in each group almost consume the same amount of energy per unit time.

Next, we describe how to assign a monitor and a parent from C_{l-1} to each sensor node in $C_l(l = m, m-1, \cdots, 2)$. Recall that for each sensor node, its parent and its monitor must satisfy the four requirements stated in the previous section.

3. Constructing A Shortest Path Overhearing Tree With Maximum Lifetime

For each sensor node $v_i \in C_l$, we use a 3-tuple $(v_i, v_j, v_k)(j < k)$ to uniquely represent a triangle formed by v_i, v_j and v_k in the connectivity graph, where v_j and v_k are in C_{l-1}.

For each sensor node $v_i \in C_l$, we introduce the following notations.

- S_i: a set of all the triangles in the connectivity graph each of which contains v_i and two sensor nodes in C_{l-1}. Notice that if S_i is empty, no shortest path overhearing tree exists.
- P_i: a set of all the sensor nodes in C_{l-1} each of which is adjacent to v_i in the connectivity graph and not selected as the monitor or the parent of v_i.
- M_i: a set of all the sensor nodes in C_{l+1} for which v_i is the monitor. Initially, M_i is $\emptyset$. Each time when v_i is selected as the monitor of a sensor node v_j in C_{l+1}, M_i is updated as follows: $M_i = M_i \cup \{v_j\}$.
- CH_i: a set of all the children of v_i in the partial shortest path tree rooted at v_i currently constructed.
- $subtree - size(v_i)$: the number of sensor nodes in the subtree rooted at the sensor node v_i constructed so far.
- e_i: the energy consumed by v_i per unit time under the current assignment of monitors and parents. Initially, e_i is equal to β. Each time when v_i is selected as the monitor of a sensor node v_j, e_i is updated as follows: $e_i = e_i + 2tree - size(v_j)\alpha$. Each time when v_i is selected as the parent of a sensor node v_j, e_i is updated as follows: $e_i = e_i + tree - size(v_j)\alpha + tree - size(v_j)\beta$.

We define two types of priorities: a priority for each sensor node in C_l and a priority for each sensor node in C_{l-1}. The priority $P_1(v_i)$ of each sensor node v_i in C_l is a 3-tuple defined as follows:

$$P_1(v_i) = (|S_i|, |P_i|, 1/tree - size(v_i)) \tag{3.1}$$

The priority $P_2(v_j)$ of each sensor node v_j in C_{l-1} is a 2-tuple defined as follows:

$$P_2(v_j) = (e_j, |M_j \cup CH_j|) \tag{3.2}$$

For both types of priorities, a smaller tuple implies a higher priority as shown in Figure 3.4. Note that the priority of each sensor node may be changed dynamically during the initial monitor and parent selection phase.

For each group $C_l(l = m, m-1, \cdots, 2)$, the initial monitor and parent selection algorithm works as follows:

1. For each sensor node v_i in C_l, compute S_i and P_i.
2. For each sensor node v_j in C_{l-1}, do the following.
   (a) Set CH_j and M_j to $\emptyset$.
   (b) Set e_j to β.
3. For each sensor node $v_i \in C_l$, compute the priority $P_1(v_i)$.
4. For each sensor node $v_j \in C_{l-1}$, compute the priority $P_2(v_j)$.
5. $A = C_l$.
6. Repeat the following until A is $\emptyset$.
   (a) Select a sensor node v_i with the highest priority from A.
   (b) If S_i is equal to $\emptyset$, no parent and monitor exist for v_i, and the algorithm terminates.

(c) Assign a 2-tuple rank $R(X)$ to each triangle $X \in S_i$ as follows:

i. Let v_j and v_k be the two sensor nodes other than v_i in X such that $P_2(v_j) \leq P_2(v_k)$ holds.

ii. $R(X) = (P_2(v_j), P_2(v_k))$.

(d) Find a triangle X_{min} in S_i with the smallest rank.

(e) Let $R(X_{min}) = (P_2(v_s), P_2(v_t))$.

(f) If $e_s < e_t$ holds, do the following.

i. Select v_s and v_t as the parent and the monitor of v_i, respectively.

ii. $M_t = M_t \cup \{v_i\}$.

iii. $CH_s = CH_s \cup \{v_i\}$.

iv. $e_s = e_s + tree - size(v_i)\alpha + tree - size(v_i)\beta$.

v. $e_t = e_t + 2tree - size(v_i)\alpha$.

vi. Re-compute $P_2(v_s)$ and $P_2(v_t)$.

Otherwise, do the following.

i. Select v_t and v_s as the parent and the monitor of v_i, respectively.

ii. $M_s = M_s \cup \{v_i\}$.

iii. $CH_t = CH_t \cup \{v_i\}$.

iv. $e_t = e_t + tree - size(v_i)\alpha + tree - size(v_i)\beta$.

v. $e_s = e_s + 2tree - size(v_i)\alpha$.

vi. Re-compute $P_2(v_s)$ and $P_2(v_t)$.

(g) $A = A - \{v_i\}$.

(h) For each sensor node $v_j \in A$ that is adjacent to v_s in the connectivity graph, set P_j to $P_j - \{v_s\}$.

(i) For each sensor node $v_j \in A$ that is adjacent to v_t in the connectivity graph, set P_j to $P_j - \{v_t\}$.

(j) For each sensor node $v_j \in A$ that is adjacent to v_s or v_t in the connectivity graph, re-compute $P_1(v_j)$.

After the initial monitor and parent selection phase, the energy balancing phase starts. The energy balancing phase works from the group C_{m-1} to the group C_1. For each group $C_l(l = m-1, m-2, \cdots, 1)$, the energy balancing algorithm selects a sensor node v_i with the maximum energy consumption per unit time, and shifts the role of v_i as the parent or the monitor of a sensor node v_j to another sensor node with lower energy consumption. In order to find such a sensor node v_j, we define a new rank for each sensor node v_r in $M_i \cup CH_i$ as follows:

$$W(v_r) = (W_1(v_r), W_2(v_r), \cdots, W_N(v_r)) \tag{3.3}$$

where N is equal to $|C_l|$, and $W_1(v_r)$, $W_2(v_r)$, $\cdots$, $W_N(v_r)$ are the energy consumptions per time unit of all the sensor nodes in C_l sorted in non-increasing order after v_i's role as v_r's monitor or parent is switched to a candidate sensor node v_p in C_l.

Let v_r be a sensor node in $M_i \cup CH_i$ such that the role of v_i as the monitor or the parent of v_r will be replaced by a candidate sensor node v_p. v_p is found as follows:

1. Let B be a set of all the triangles of the form $(v_r, v_i, v_s)(i < s)$ or $(v_r, v_s, v_i)(s < i)$.
2. Let E be a set of all the sensor nodes in B that are different from v_r and v_i.
3. v_p is the sensor node in E that has the smallest energy consumption per time unit.

If such a sensor node v_p is not found, $W(v_r)$ is set to $(+\infty, +\infty, \cdots, +\infty)$. A role switch is allowed only if the switch results in better energy balancing.

The energy balancing algorithm works for each group $C_l(l = m-1, m-2, \cdots, 1)$ as follows.

3. Constructing A Shortest Path Overhearing Tree With Maximum Lifetime

1. Mark each sensor node in C_l as switchable.
2. Repeat the following until no sensor node in C_l is switchable.
   (a) Let $e_1, e_2, \cdots, e_N$ be the energy consumptions per time unit of all the sensor nodes in C_l sorted in non-increasing order.
   (b) $W = (e_1, e_2, \cdots, e_N)$.
   (c) Pick a sensor node v_i in C_l that is not marked as unswitchable and has the maximum energy consumption per unit time.
   (d) For each sensor node $v_s \in M_i \cup CH_i$, compute $W(v_s)$.
   (e) $T = \min\{W(v_s) : v_s \in M_i \cup CH_i\}$.
   (f) If $T \geq W$ holds, mark v_i as unswitchable.
   (g) Otherwise, let v_r be a sensor node in $M_i \cup CH_i$ with the smallest rank $W(v_r)$, and v_p the sensor node selected to replace v_i's role as the parent or the monitor of v_r when computing $W(v_r)$. Do the following.
      i. Switch v_i's role as v_r's monitor or parent to v_p.
      ii. Re-compute e_i and e_p.
3. If $l > 1$ holds, for each sensor node v_j in C_{l-1}, re-calculate e_j based on the current partial shortest path overhearing tree.

Next, we analyse the time complexity of our heuristic approach. The time complexity is broken down into the following three parts.

1. The partitioning phase. We can use breadth-first search to compute the shortest path length from each sensor node to the base station. Therefore, the time complexity of this phase is $O(e)$, where e is the number of edges in the connectivity graph.
2. The initial monitor and parent selection phase. First, we assume that for each sensor node, the maximum number of sensor nodes it can communicate directly is

a constant. Under this assumption, for each sensor node v_i, $O(|S_i|) = O(|P_i|) = O(|M_i \cup CH_i|) = O(1)$ holds. The breakdown of the time complexity of this phase is as follows.

(a) Steps 1-5. Notice that the connectivity graph is connected. Therefore, the time complexity of steps 1-5 for all the groups is $O(e)$.

(b) Step 6. The time complexity of this step for all the groups is $O(ne)$, where n is the number of sensor nodes in the WSN.

As a result, the time complexity of this phase is $O(ne)$.

3. The energy balancing phase. Notice that each time when a role switch occurs, W will decrease monotonically. For each sensor node v_i in C_l, it is processed at most $|CH_i|$ times by the energy balancing algorithm. Therefore, the energy balancing algorithm terminates in $\sum_{v_i \in C_l}(|CH_i| = O(|C_l|)$ steps for each group C_l. For each group C_l, each step takes $O(|C_l| \log |C_l|)$. As a result, the time complexity of this phase is $O(n^2 \log n)$.

As discussed above, the time complexity of our heuristic approach is $O(e) + O(ne) + O(n^2 \log n) = O(ne + n^2 \log n)$.

3.4.2 A Comprehensive Example

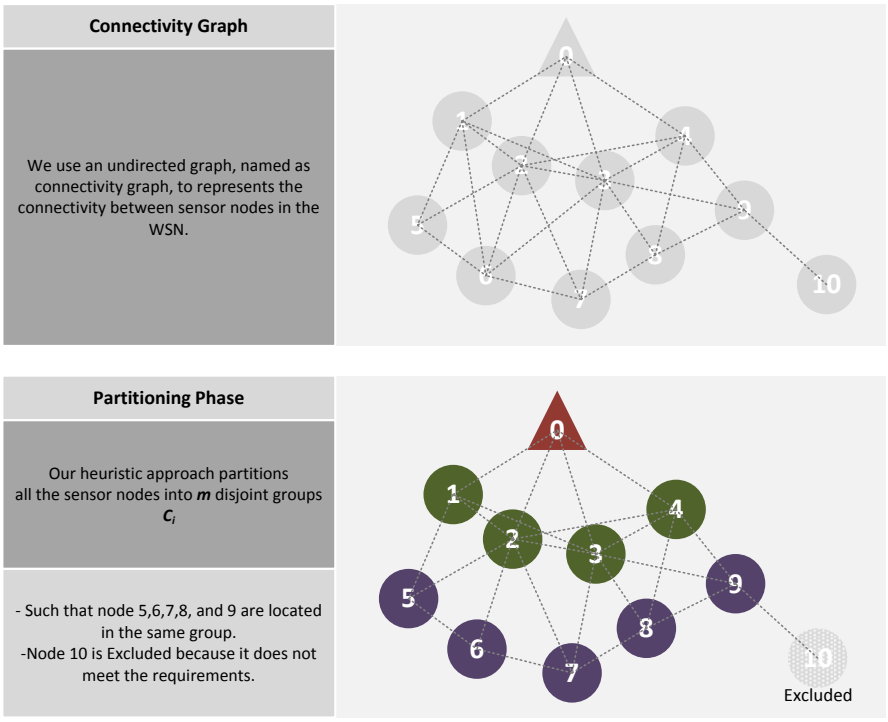

Figure 3.1: Partitioning Phase

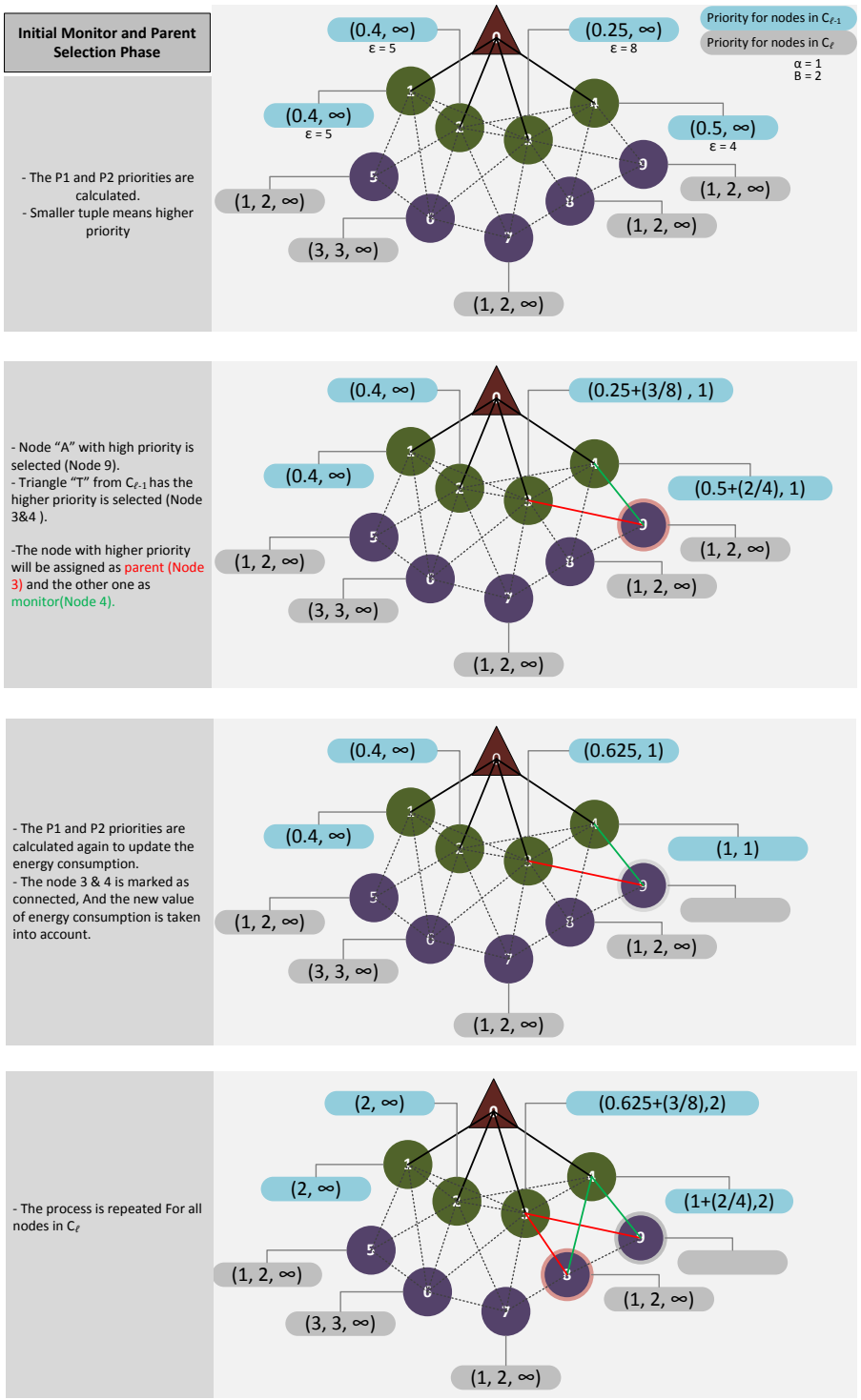

Figure 3.2: Initial Parent and Monitor Assignment

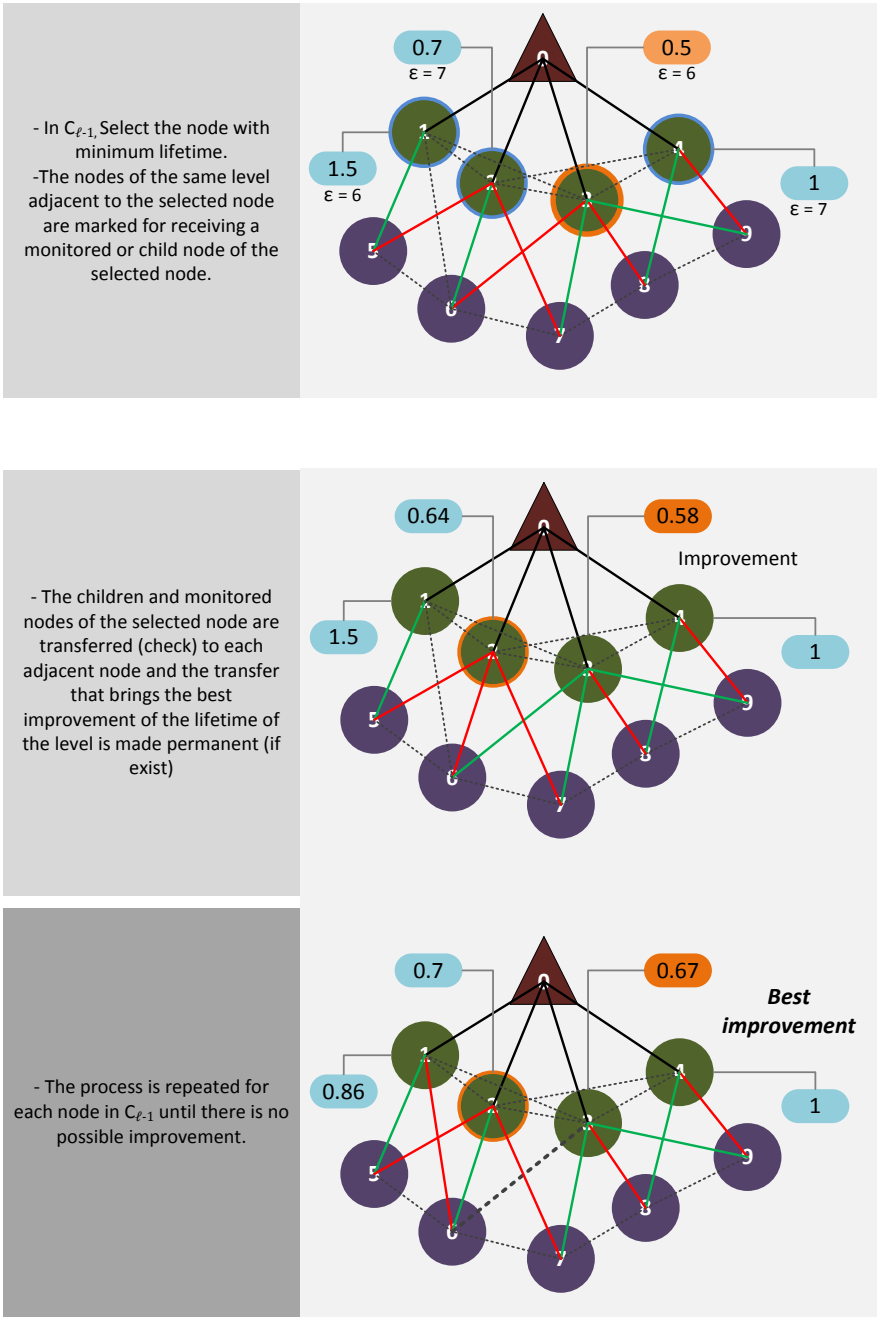

Figure 3.3: Adjustment Phase

Priority Calculation for ($P_1(6)$)

$(|S_6|, |P_6|, 1/((tree\text{-}size(v_i)))$

$|S6|$: Triangles where node 6 belongs
In this case $|S6|$ is 3

$|P6|$: Nodes in $C_{\ell-1}$ reaching node 6
In this case $|P6|$ is 3

($(tree\text{-}size(v_6)$):Tree size connecting to Node 6 through Parent links
In this case **($(tree\text{-}size(v_6)$)** *is 2*

$P_1(6)=(3, 3, 0.5)$

Priority Calculation for ($P_2(2)$)

$(1/\tau_2, |CH_2 \cup M_2|)$

e_2 : Energy consumed by Node 2 per unit time, initially β

ε_2 : Initial total energy of Node 2

$\tau_2 = \varepsilon_2/e_2$: Energy consumed by Node 2 per unit time.

M_2 : a set of all the sensor nodes in $C\ell+1$ or which v_i is the monitor.
In this case M_2 is 2.

Each time node 2 is selected as a Monitor for Node "j" located in C_ℓ
$e_2 = e_2 + 2\alpha((tree\text{-}size(v_2)+1)$
In this case:
$\alpha = 1$; $\beta = 2$; $\varepsilon = 5$;
$e_2 = 2$
$e_2 = e_2 + 2\alpha((tree\text{-}size(v_5)+1) = e_2 + 2\alpha(1+1)$
$e_2 = 6$
$e_2 = e_2 + 2\alpha((tree\text{-}size(v_7)+1) = e_2 + 2\alpha(3+1)$
$e_2 = 14$

Each time node 2 is selected as the Parent for Node "j" located in C_ℓ
$e_2 = e_2 + (\alpha + \beta)(tree\text{-}size(v_2))$

In this case:
$e_2 = 14$ (current energy consumed)
$e_2 = e_2 + (\alpha + \beta)(tree\text{-}size(v_2)) = e_2 + 3(3)$
$e_2 = 14+9=23$

$P_2(2)=(23/5, (3+2)) = (3.8, 5)$

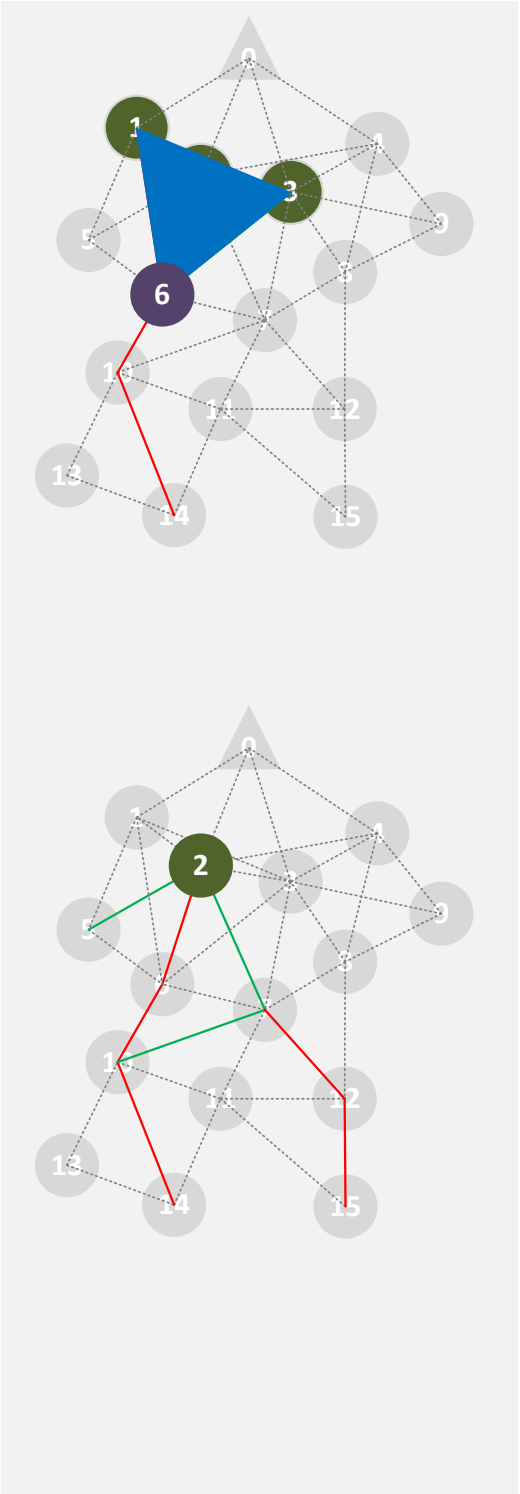

Figure 3.4: Priority Calculation/Update

3.4.3 MINLP-Based Approach

The objective of MINLP-based approach is to construct a shortest path tree for routing and assign a monitor for each sensor node such that the network lifetime is maximized while the four requirements described in Section 3.3 are met.

The MIINLP-based approach consists of two phases. In the first phase, it partitions all the sensor nodes into m disjoint groups $C_l(l = 1, 2, \cdots, m)$ such that the shortest path length between each sensor node in the group C_l to the base station in the connectivity graph is equal to l. In the second phase, for each sensor node $v_i \in C_l(l = 2, 3, \cdots, m)$, it assigns a monitor and a parent in C_{l-1} to v_i such that the maximum energy consumption per unit time of all the individual sensor nodes is minimized by using MINLP.

As in our heuristic approach, we use $(v_i, v_j, v_k)(j < k)$ to uniquely represent a triangle formed by v_i, v_j and v_k in the connectivity graph, where v_i is in C_l, v_j and v_k are in C_{l-1}. For each sensor node $v_i \in C_l$, we also use S_i to denote a set of all the triangles in the connectivity graph each of which contains v_i and two sensor nodes in C_{l-1} as in Section 3.3.

For each triangle (v_i, v_j, v_k), we introduce two binary decision variables $x_{(i,j,k)}$ and $y_{(i,j,k)}$ as follows:

$$x_{(i,j,k)} = \begin{cases} 1 & v_j \text{ is the monitor \& } v_k \text{ is the parent of } v_i \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

$$y_{(i,j,k)} = \begin{cases} 1 & v_k \text{ is the monitor \& } v_j \text{ is the parent of } v_i \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

Therefore, for each sensor node $v_i \in C_l(l = 1, 2, \cdots, m)$, we have the following monitor and parent selection constraint:

$$\sum_{(v_i,v_j,v_k)\in S_i} x_{(i,j,k)} + y_{(i,j,k)} = 1 \tag{3.6}$$

The above constraint implies that among all the sensor nodes in C_{l-1} with which v_i can communicate directly, only two sensor nodes are selected as the monitor and the parent of v_i, respectively, and the monitor and the parent can communicate with each other.

Next, we derive the energy constraint and other related constraints for each sensor node. For each sensor $v_i \in C_l(l = 1, 2, \cdots, m)$, we further introduce the following notations:

1. l_i: the number of packets v_i receives from all its children in the shortest path tree per unit time.

2. m_i: the total energy consumption per time unit by v_i being a monitor.

3. p_i: the total energy consumption per time unit by v_i being a parent.

4. e_i: the total energy consumption per unit time by v_i.

For each sensor v_i, if it is in C_m, it is a leaf node in the shortest path tree. Therefore, we have the following constraint on l_i:

$$l_i = \begin{cases} 0 & v_i \text{ is in } C_m \\ \sum_{v_k \in CH_i} \sum_{(v_k, v_i, v_j) \in S_k} y_{(k,i,j)} * l_k + & \\ \sum_{v_k \in CH_i} \sum_{(v_k, v_j, v_i) \in S_k} x_{(k,j,i)} * l_k & \text{otherwise} \end{cases} \tag{3.7}$$

If a sensor node v_i is the monitor of a sensor node v_k, it needs to overhear the packet transmission when v_k sends its packet to its parent v_j and when v_j forwards the packet to

its own parent. Therefore, we have the following constraint on m_i:

$$m_i = \begin{array}{l} \sum\limits_{v_k \in M_i} \sum\limits_{(v_k,v_i,v_j) \in S_k} x_{(k,i,j)} * 2(l_k+1)\alpha + \\ \sum\limits_{v_k \in M_i} \sum\limits_{(v_k,v_j,v_i) \in S_k} y_{(k,j,i)} * 2(l_k+1)\alpha \end{array} \tag{3.8}$$

If a sensor node v_i is a parent of a sensor node v_k, it needs to not only receive the data from v_k, but also forward the data to its parent. Therefore, we have the following constraint on p_i:

$$p_i = \begin{array}{l} \sum\limits_{v_k \in CH_i} \sum\limits_{(v_k,v_i,v_j) \in S_k} y_{(k,i,j)} * l_k(\alpha+\beta) + \\ \sum\limits_{v_k \in CH_i} \sum\limits_{(v_k,v_j,v_i) \in S_k} x_{(k,j,i)} * l_k(\alpha+\beta) \end{array} \tag{3.9}$$

For each sensor node v_i, its energy consumption per time unit consists of three parts: m_i, p_i and the energy for transmitting its own packet. Therefore, we have the following constraint on e_i:

$$e_i = m_i + p_i + \beta \tag{3.10}$$

Our optimization objective function is as follows:

$$\min \max_{v_i \in V} \{e_i\} \tag{3.11}$$

3.4.4 ILP-Based Approach

The ILP-based approach consists of two phases. In the first phase, it partitions all the sensor nodes into m disjoint groups $C_l (l = 1, 2, \cdots, m)$ such that the shortest path length between each sensor node in the group C_l to the base station in the connectivity graph is

equal to l. In the second phase, for each group $C_l(l = m, m-1, \cdots, 2)$, it assigns a monitor and a parent in C_{l-1} to each sensor in C_l such that the maximum energy consumption per time unit of all the individual sensor nodes in C_{l-1} is minimized by using ILP.

Next, we show how to use ILP to find a locally optimal assignment of a monitor and a parent for each sensor node in $C_l(l = m, m-1, \cdots, 2)$.

Similar to the MINLP approach, for each sensor node $v_i \in C_l$, we have the following monitor and parent selection constraint:

$$\sum_{(v_i,v_j,v_k)\in S_i} x_{(i,j,k)} + y_{(i,j,k)} = 1 \tag{3.12}$$

The binary decision variables $x_{(i,j,k)}$ and $y_{(i,j,k)}$ are defined in the same way as in the MINLP approach.

For each sensor node $v_i \in C_{l-1}$, we have the following energy constraint on m_i:

$$m_i = \begin{array}{l} \sum_{v_k\in M_i} \sum_{(v_k,v_i,v_j)\in S_k} x_{(k,i,j)} * 2(l_k+1)\alpha + \\ \sum_{v_k\in M_i} \sum_{(v_k,v_j,v_i)\in S_k} y_{(k,j,i)} * 2(l_k+1)\alpha \end{array} \tag{3.13}$$

If v_k is a sensor node in C_m, l_k is equal to 0. Notice that each l_k is a constant as our ILP-based approach has finished the monitor and parent assignment for each group $C_j(j > l)$.

For each sensor node $v_i \in C_{l-1}$, we have the following constraint on p_i:

$$p_i = \begin{array}{l} \sum_{v_k\in CH_i} \sum_{(v_k,v_i,v_j)\in S_k} y_{(k,i,j)} * l_k(\alpha+\beta) + \\ \sum_{v_k\in CH_i} \sum_{(v_k,v_j,v_i)\in S_k} x_{(k,j,i)} * l_k(\alpha+\beta) \end{array} \tag{3.14}$$

For each sensor node $v_i \in C_{l-1}$, we have the following constraint on e_i:

$$e_i = m_i + p_i + \beta \tag{3.15}$$

Our optimization objective function is as follows:

$$\min \max_{v_i \in C_{l-1}} \{e_i\} \tag{3.16}$$

After selecting the parent and the monitor of each sensor node in C_l, our ILP-based approach computes l_k for each sensor node v_k in C_{l-1}.

3.5 Constructing A Shortest Path Overhearing Tree With Maximum Lifetime In Heterogeneous WSNs

In this section we show an extension to the aforementioned algorithms to be applied for a heterogeneous network. There are two major differences between heterogeneous WSNs and homogeneous WSNs.

Firstly, all the sensor nodes have the same initial energy in homogeneous WSNs while each sensor node may have a different initial energy in heterogeneous WSNs. Secondly, all the sensor nodes has the same α and the same β in homogeneous WSNs while each sensor node has its own α and β in homogeneous WSNs.

3.5.1 Heuristic Approach for Heterogeneous WSNs

We modify the priority $P_2(v_j)$ of each sensor node v_j in C_k as follows:

$$P_2(v_j) = (k, \frac{1}{E_j}, |M_j \cup CH_j|) \tag{3.17}$$

3. Constructing A Shortest Path Overhearing Tree With Maximum Lifetime

We extend our previous heuristic approach as follows:

1. For each sensor node v_i in C_l, compute S_i and P_i.
2. For each triangle in S_i set a parameter k and initialize it by 0.
3. For each sensor node v_j in C_{l-1}, do the following.
   (a) Set CH_j and M_j to $\emptyset$.
   (b) Set e_j to β_i.
4. Set a timer t.
5. For each sensor node $v_i \in C_l$, compute the priority $P_1(v_i)$.
6. For each sensor node $v_j \in C_{l-1}$, compute the priority $P_2(v_j)$.
7. $A = C_l$.
8. Repeat the following until A is $\emptyset$.
   (a) Select a sensor node v_i with the highest priority from A.
   (b) If S_i is equal to $\emptyset$, no parent and monitor exist for v_i, and the algorithm terminates.
   (c) Assign a 2-tuple rank $R(X)$ to each triangle $X \in S_i$ as follows:
      i. Let v_j and v_k be the two sensor nodes other than v_i in X such that $P_2(v_j) \leq P_2(v_k)$ holds.
      ii. $R(X) = (P_2(v_j), P_2(v_k))$.
   (d) Find a triangle X_{min} in S_i with the smallest rank.
   (e) Let $R(X_{min}) = (P_2(v_s), P_2(v_t))$.
   (f) If $\frac{1}{\tau_s} < \frac{1}{\tau_t}$ holds, do the following.
      i. Select v_s and v_t as the parent and the monitor of v_i, respectively.

ii. $M_t = M_t \cup \{v_i\}$.

iii. $CH_s = CH_s \cup \{v_i\}$.

iv. $e_s = e_s + tree - size(v_i)\alpha + tree - size(v_i)\beta$.

v. $e_t = e_t + 2tree - size(v_i)\alpha$.

vi. Re-compute $P_2(v_s)$ and $P_2(v_t)$.

vii. Increase k by one for the selected triangle.

viii. Once $t = 0$, go to 4.

Otherwise, do the following.

i. Select v_t and v_s as the parent and the monitor of v_i, respectively.

ii. $M_s = M_s \cup \{v_i\}$.

iii. $CH_t = CH_t \cup \{v_i\}$.

iv. $e_t = e_t + tree - size(v_i)\alpha + tree - size(v_i)\beta$.

v. $e_s = e_s + 2tree - size(v_i)\alpha$.

vi. Re-compute $P_2(v_s)$ and $P_2(v_t)$.

vii. Increase k by one for the selected triangle.

viii. Once $t = 0$, go to 4.

(g) $A = A - \{v_i\}$.

(h) For each sensor node $v_j \in A$ that is adjacent to v_s in the connectivity graph, set P_j to $P_j - \{v_s\}$.

(i) For each sensor node $v_j \in A$ that is adjacent to v_t in the connectivity graph, set P_j to $P_j - \{v_t\}$.

(j) For each sensor node $v_j \in A$ that is adjacent to v_s or v_t in the connectivity graph, re-compute $P_1(v_j)$.

3.5.2 MINLP Approach for Heterogeneous WSNs

The objective of MINLP-based approach is to construct a shortest path tree for routing and assign a monitor for each sensor node such that the network lifetime is maximized while the four requirements described in the network model section are met. We have updated our formulation for the purpose of heterogeneous WSN.

As previously explained, the MIINLP-based approach consists of two phases. In the first phase, it partitions all the sensor nodes into m disjoint groups $C_l(l = 1, 2, \cdots, m)$ such that the shortest path length between each sensor node in the group C_l to the base station in the connectivity graph is equal to l. In the second phase, for each sensor node $v_i \in C_l(l = 2, 3, \cdots, m)$, it assigns a monitor and a parent in C_{l-1} to v_i such that the minimum lifetime per unit time of all the individual sensor nodes is maximized by using MINLP.

As in our heuristic approach, we use $(v_i, v_j, v_k)(j < k)$ to uniquely represent a triangle formed by v_i, v_j and v_k in the connectivity graph, where v_i is in C_l, v_j and v_k are in C_{l-1}. For each sensor node $v_i \in C_l$, we also use S_i to denote a set of all the triangles in the connectivity graph each of which contains v_i and two sensor nodes in C_{l-1} as in Section 3.3.

For each triangle (v_i, v_j, v_k), we introduce two binary decision variables $x_{(i,j,k)}$ and $y_{(i,j,k)}$ as follows:

$$x_{(i,j,k)} = \begin{cases} 1 & v_j \text{ is the monitor \& } v_k \text{ is the parent of } v_i \\ 0 & \text{otherwise} \end{cases} \tag{3.18}$$

$$y_{(i,j,k)} = \begin{cases} 1 & v_k \text{ is the monitor \& } v_j \text{ is the parent of } v_i \\ 0 & \text{otherwise} \end{cases} \tag{3.19}$$

Therefore, for each sensor node $v_i \in C_l(l = 1, 2, \cdots, m)$, we have the following monitor and parent selection constraint:

$$\sum_{(v_i,v_j,v_k)\in S_i} x_{(i,j,k)} + y_{(i,j,k)} = 1 \tag{3.20}$$

The above constraint implies that among all the sensor nodes in C_{l-1} with which v_i can communicate directly, only two sensor nodes are selected as the monitor and the parent of v_i, respectively, and the monitor and the parent can communicate with each other.

Next, we derive the energy constraint and other related constraints for each sensor node. For each sensor $v_i \in C_l(l = 1, 2, \cdots, m)$, we further introduce the following notations:

1. l_i: the number of packets v_i receives from all its children in the shortest path tree per unit time.

2. m_i: the total energy consumption per time unit by v_i being a monitor.

3. p_i: the total energy consumption per time unit by v_i being a parent.

4. e_i: the total energy consumption per unit time by v_i.

For each sensor v_i, if it is in C_m, it is a leaf node in the shortest path tree. Therefore, we have the following constraint on l_i:

$$l_i = \begin{cases} 0 & v_i \text{ is in } C_m \\ \sum_{v_k\in CH_i} \sum_{(v_k,v_i,v_j)\in S_k} y_{(k,i,j)} * l_k + & \\ \sum_{v_k\in CH_i} \sum_{(v_k,v_j,v_i)\in S_k} x_{(k,j,i)} * l_k & \text{otherwise} \end{cases} \tag{3.21}$$

3. Constructing A Shortest Path Overhearing Tree With Maximum Lifetime

If a sensor node v_i is the monitor of a sensor node v_k, it needs to overhear the packet transmission when v_k sends its packet to its parent v_j and when v_j forwards the packet to its own parent. Therefore, we have the following constraint on m_i:

$$m_i = \begin{array}{l} \sum\limits_{v_k \in M_i} \sum\limits_{(v_k,v_i,v_j) \in S_k} x_{(k,i,j)} * 2(l_k+1)\alpha_i + \\ \sum\limits_{v_k \in M_i} \sum\limits_{(v_k,v_j,v_i) \in S_k} y_{(k,j,i)} * 2(l_k+1)\alpha_i \end{array} \tag{3.22}$$

If a sensor node v_i is a parent of a sensor node v_k, it needs to not only receive the data from v_k, but also forward the data to its parent. Therefore, we have the following constraint on p_i:

$$p_i = \begin{array}{l} \sum\limits_{v_k \in CH_i} \sum\limits_{(v_k,v_i,v_j) \in S_k} y_{(k,i,j)} * l_k(\alpha_i + \beta_i) + \\ \sum\limits_{v_k \in CH_i} \sum\limits_{(v_k,v_j,v_i) \in S_k} x_{(k,j,i)} * l_k(\alpha_i + \beta_i) \end{array} \tag{3.23}$$

For each sensor node v_i, its energy consumption per time unit consists of three parts: m_i, p_i and the energy for transmitting its own packet. Therefore, we have the following constraint on e_i:

$$e_i = m_i + p_i + \beta_i \tag{3.24}$$

Our optimization objective function becomes:

$$\min \max_{v_i \in V} (\frac{e_i}{E_i}) \tag{3.25}$$

3.5.3 ILP Approach for Heterogeneous WSNs

The ILP-based approach consists of two phases. In the first phase, it partitions all the sensor nodes into m disjoint groups $C_l(l = 1, 2, \cdots, m)$ such that the shortest path length between each sensor node in the group C_l to the base station in the connectivity graph is equal to l. In the second phase, for each group $C_l(l = m, m-1, \cdots, 2)$, it assigns a monitor and a parent in C_{l-1} to each sensor in C_l such that the minimum lifetime per time unit of all the individual sensor nodes in C_{l-1} is maximized by using ILP.

Next, we show how to use ILP to find a locally optimal assignment of a monitor and a parent for each sensor node in $C_l(l = m, m-1, \cdots, 2)$.

Similar to the MINLP approach, for each sensor node $v_i \in C_l$, we have the following monitor and parent selection constraint:

$$\sum_{(v_i,v_j,v_k)\in S_i} x_{(i,j,k)} + y_{(i,j,k)} = 1 \tag{3.26}$$

The binary decision variables $x_{(i,j,k)}$ and $y_{(i,j,k)}$ are defined in the same way as in the MINLP approach.

For each sensor node $v_i \in C_{l-1}$, we have the following energy constraint on m_i:

$$m_i = \begin{array}{l} \displaystyle\sum_{v_k\in M_i} \sum_{(v_k,v_i,v_j)\in S_k} x_{(k,i,j)} * 2(l_k+1)\alpha_i + \\ \displaystyle\sum_{v_k\in M_i} \sum_{(v_k,v_j,v_i)\in S_k} y_{(k,j,i)} * 2(l_k+1)\alpha_i \end{array} \tag{3.27}$$

If v_k is a sensor node in C_m, l_k is equal to 0. Notice that each l_k is a constant as our ILP-based approach has finished the monitor and parent assignment for each group $C_j(j > l)$.

For each sensor node $v_i \in C_{l-1}$, we have the following constraint on p_i:

$$p_i = \frac{\sum_{v_k \in CH_i} \sum_{(v_k,v_i,v_j) \in S_k} y_{(k,i,j)} * l_k(\alpha_i + \beta_i) +}{\sum_{v_k \in CH_i} \sum_{(v_k,v_j,v_i) \in S_k} x_{(k,j,i)} * l_k(\alpha_i + \beta_i)} \tag{3.28}$$

For each sensor node $v_i \in C_{l-1}$, we have the following constraint on e_i:

$$e_i = m_i + p_i + \beta_i \tag{3.29}$$

Our optimization objective function becomes:

$$\min \max_{v_i \in C_{l-1}} (\frac{e_i}{E_i}) \tag{3.30}$$

After selecting the parent and the monitor of each sensor node in C_l, our ILP-based approach computes l_k for each sensor node v_k in C_{l-1}.

3.6 Discussion

One limitation with our shortest path overhearing tree is that if a monitor colludes with the parent of the monitored sensor node, the attacks by the parent of the monitored sensor node may not be detected.

In this section, we provide a more secure shortest path overhearing tree where there are multiple monitors for each sensor node, such that we compute a set of the available triangles which can be performed for the child node v_i, and we intend to exchange between triangles in every time interval to provide a different monitor for every child node.

Accordingly, our heuristic approach is modified as follows:

3. Constructing A Shortest Path Overhearing Tree With Maximum Lifetime

- In every particular time interval t, sort the set of the available triangles for each sensor node v_i based on their lifetimes avoid selecting a previously selected triangle unless no choices is left.
- Add one more parameter k to indicate how many times a triangle has been selected. k is initialized by 0 and increases by 1 after each selection.

We update P_2 for each sensor v_j as follows:

$$P_2(v_j) = (k, \frac{1}{E_j}, |M_j \cup CH_j|) \tag{3.31}$$

In order to make our MINLP-Based approach and IL-Based approach satisfy this new requirement, we introduce a new set A_i as follows:

- A_i: is a set of all selected triangles in S_i each of which contains v_i and two sensor nodes in C_{l-1}. Notice that if A_i is empty, either no shortest path tree exist or no triangles has been selected yet.

We replace each S_i in the constraints of our MINLP-Based approach and IL-Based approach by $S_i - A_i$.

3.7 Simulation Results

3.7.1 Setup

In order to evaluate our approaches, we use NS 2.35 to generate 150 instances with three different distributions, namely, uniform, grid, and random distributions. NS 2.35 provides a lot of parameters for modelling a real wireless sensor network and thus has been widely used by researchers. In grid distribution, a WSN consists of lines of sensor nodes that cross

each other to form a series of squares or rectangles. In uniform distribution, sensor nodes are deployed uniformly with the same density across the network. In random distribution, sensor nodes are deployed at random in the network. We vary the number of sensor nodes from 100 to 300 with an increment of 50. For each scenario with a fixed number of sensor nodes and a particular distribution, we generate 10 instances. For each instance, sensor nodes are deployed in a 300 m x 400 m rectangular area, and the base station is deployed at the middle of the upper boundary of the rectangular area. MIDACO Solver is used to solve the MINLP problems and the *Intlinprog* Solver of MATLAB is used for the ILP problems.

For homogeneous network, the initial energy of every sensor node is 0.5 KJ. The transmission range is fixed to 70 m. The energy consumption for receiving one packet of data is $\alpha = 0.001$ KJ while the energy consumed to transmit one packet of data is $\beta = 0.002$ KJ.

For heterogeneous network, the initial energy of every sensor node is bounded by 0.4 J to 0.5 J. The transmission range is fixed to 70 m. The energy consumption for receiving one bit of data is bounded by $\alpha = 0.234$ μJ to 0.312 μJ while the energy consumed to transmit one bit of data is bounded by $\beta = 0.234$ μJ to 0.312 μJ.

The hardware platform is Intel Core i5-3470 with a clock frequency of 3.20 Ghz, a memory size of 8 GB and a cache size of 8134 MB.

3.7.2 Homogeneous Network Simulation Results

Figures 3.5, 3.6, and 3.7 show the network lifetimes of the shortest path overhearing trees constructed by the three approaches for all the instances in the three different distributions.

The simulation results show that the heuristic approach obtains comparable network lifetimes compared with the ILP-based approach and the MINLP-based approach. For ex-

ample, for the instance having 150 sensor nodes with uniform distribution as shown in Figure 3.6, the network lifetime obtained by the heuristic approach is 3.4, comparable to the network lifetime 4.42 obtained by the ILP-based approach, and the network lifetime 4.5 achieved by the MINLP-based approach.

The relative maximum network lifetime obtained by the heuristic approach occurs in the instance of 100 randomly distributed sensor nodes, as shown in Figure 3.7, and the network lifetime is 7.29. The relative minimum network lifetime obtained by the heuristic approach is 3.4, which occurs in the instance of 150 uniformly distributed nodes, as shown in Figure 3.6.

Comparing the heuristic approach with the MINLP-based approach, the minimum ratio between the network lifetime achieved by the heuristic approach and the network lifetime obtained by the MINLP-based approach is 76.6%, which occurs in the instance of 150 nodes as shown in Figure 3.6 while the maximum ratio is 85.62% as shown in Figure 3.6. The average ratios for the grid distribution, the uniform distribution and the random distribution are 83.68%, 81.13%, and 78.25%, respectively. The average ratio for all the instances with the three different distributions is 81.05%.

In comparison between the ILP-based approach and the MINLP-based approach, the minimum ratio between the network lifetime achieved by the ILP-based approach and the network lifetime obtained by the MINLP-based approach is 94.43%. The maximum ratio is 98.24%. The average ratios for the grid distribution, the uniform distribution and the random distribution are 94.43%, 98.23%, and 93.7%, respectively. The average ratio for all the instances with the three different distributions is 96.2%.

In comparison between the heuristic approach and the ILP-based approach, the minimum ratio between the network lifetime achieved by the heuristic approach and the network lifetime obtained by the ILP-based approach is 78.04% while the maximum ratio is 93.1%. The average ratios for the grid distribution, the uniform distribution and the random

distribution are 85.6%, 83.94%, and 87.5%, respectively. The average ratio for all the instances with the three different distributions is 85.69%.

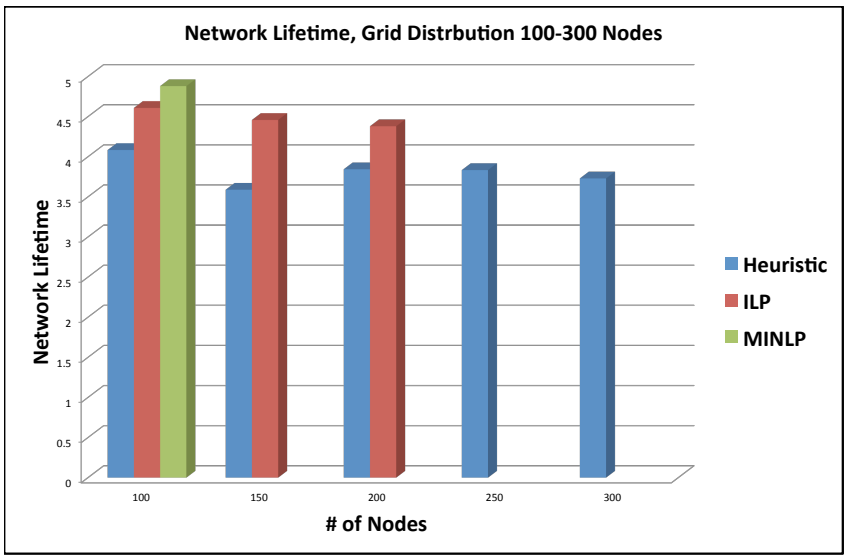

Figure 3.5: Network Lifetimes For Grid Distribution

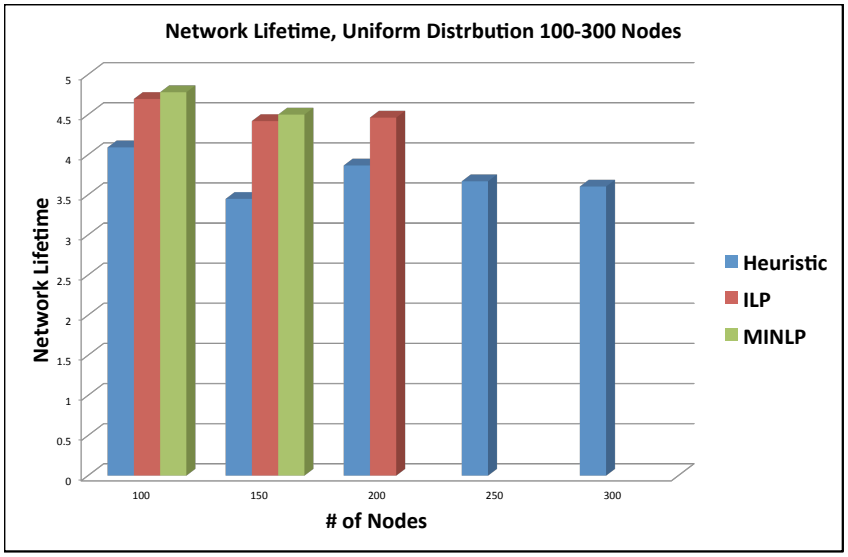

Figure 3.6: Network Lifetimes For Uniform Distribution

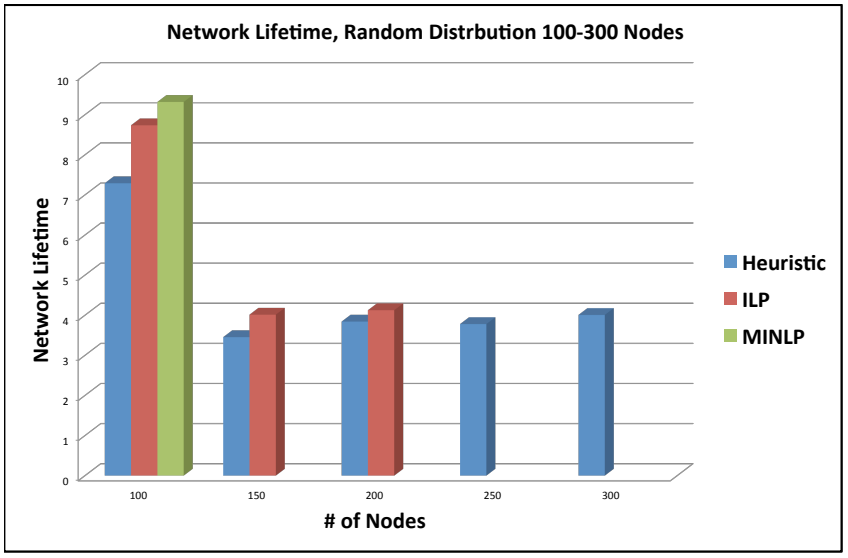

Figure 3.7: Network Lifetimes For Random Distribution

Figures 3.8, 3.9,and 3.10 shows the running times of all the three approaches for all the instances with the three different distributions.

The simulation results show that the heuristic approach always constructs a shortest path overhearing tree in a reasonable amount of time for all the instances while both the MINLP-based approach and the ILP-based approach do not scale. For example, the ILP-based approach fails to construct a shortest path overhearing tree for the instances with more than 200 sensor nodes in 3 hours while the MINLP-based approach fails to construct a shortest path overhearing tree for the instances with more than 100 sensor nodes in 3 days.

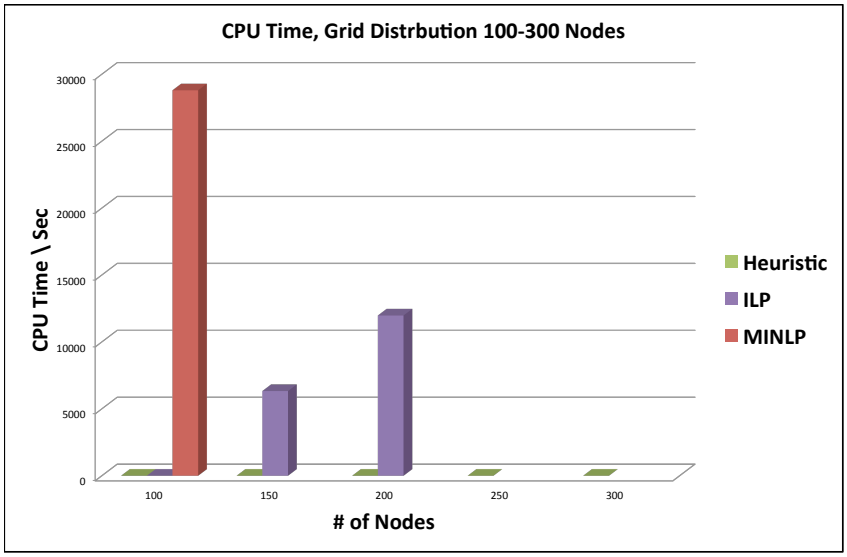

Figure 3.8: Running Times For Grid Distribution

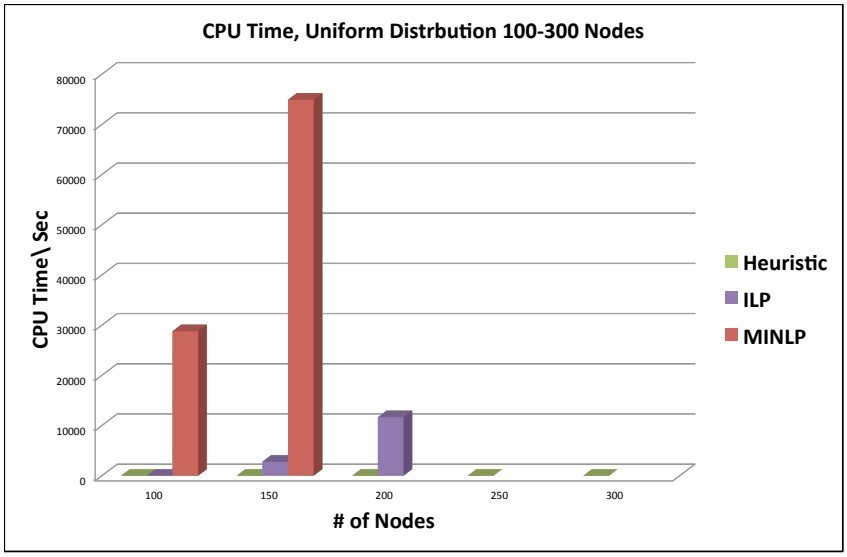

Figure 3.9: Running Times For Uniform Distribution

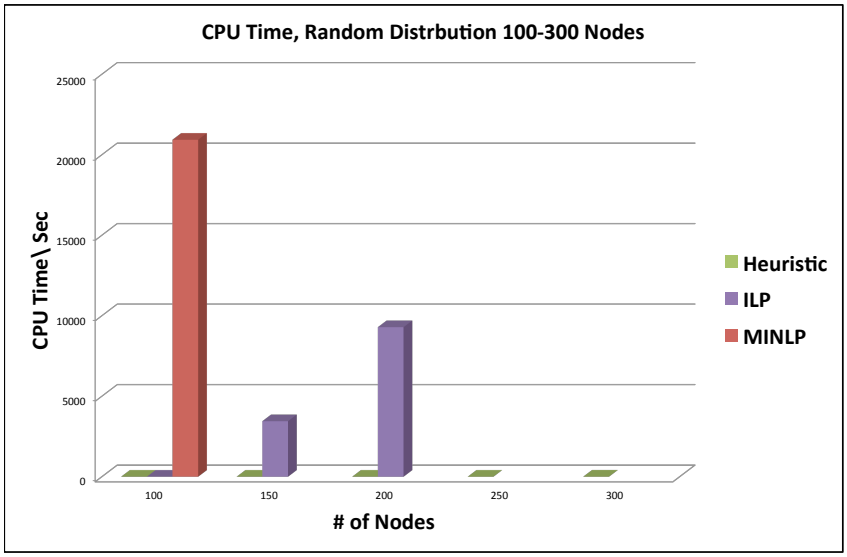

Figure 3.10: Running Times For Random Distribution

3.7.3 Heterogeneous Network Simulation Results

Figures 3.11, 3.12, and 3.13 show the network lifetimes of the shortest path overhearing trees constructed by the three approaches for all the instances in the three different distributions.

The simulation results show that the heuristic approach obtains comparable network lifetimes compared with the ILP-based approach and the MINLP-based approach. For example, for the instance having 100 sensor nodes with uniform distribution as shown in Figure 3.12, the network lifetime obtained by the heuristic approach is 1.903, comparable to the network lifetime 2.039 obtained by the ILP-based approach, and the network lifetime 2.4135 achieved by the MINLP-based approach.

The relative maximum network lifetime obtained by the heuristic approach occurs in the instance of 100 randomly distributed sensor nodes, as shown in Figure 3.13, and the network lifetime is 3.99. The relative minimum network lifetime obtained by the heuristic approach is 1.55, which occurs in the instance of 200 random distributed nodes, as shown in Figure 3.13.

Comparing the heuristic approach with the MINLP-based approach, the minimum ratio between the network lifetime achieved by the heuristic approach and the network lifetime obtained by the MINLP-based approach is 72.83%, which occurs in the instance of 100 nodes as shown in Figure 3.11 while the maximum ratio is 78.83% as shown in Figure 3.12. The average ratios for the grid distribution, the uniform distribution and the random distribution are 72.83%, 78.829%, and 73.048%, respectively. The average ratio for all the instances with the three different distributions is 74.9%.

In comparison between the ILP-based approach and the MINLP-based approach, the minimum ratio between the network lifetime achieved by the ILP-based approach and the network lifetime obtained by the MINLP-based approach is 79.83%. The maximum ratio

is 84.51%. The average ratios for the grid distribution, the uniform distribution and the random distribution are 81.058%, 84.51%, and 79.83%, respectively. The average ratio for all the instances with the three different distributions is 81.799%.

In comparison between the heuristic approach and the ILP-based approach, the minimum ratio between the network lifetime achieved by the heuristic approach and the network lifetime obtained by the ILP-based approach is 82.115% while the maximum ratio is 93.281%. The average ratios for the grid distribution, the uniform distribution and the random distribution are 86.3%, 87.775%, and 88.01%, respectively. The average ratio for all the instances with the three different distributions is 87.371%.

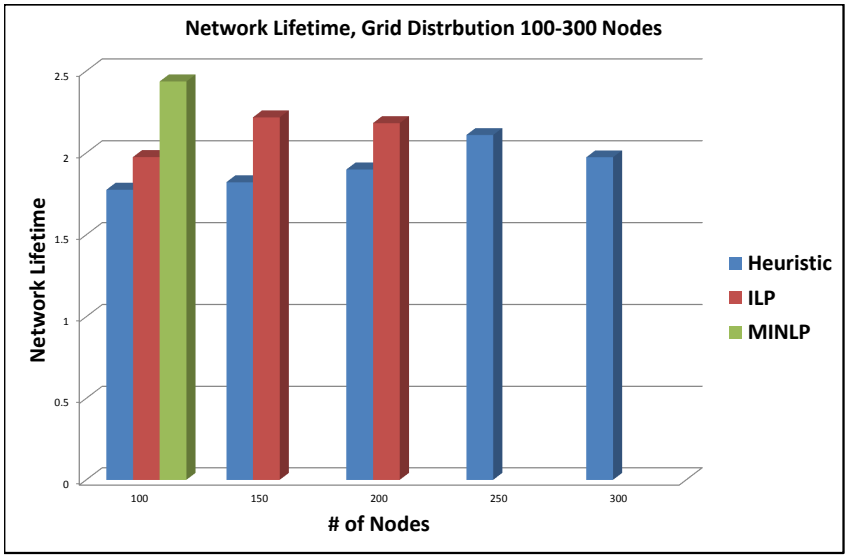

Figure 3.11: Network Lifetimes For Grid Distribution

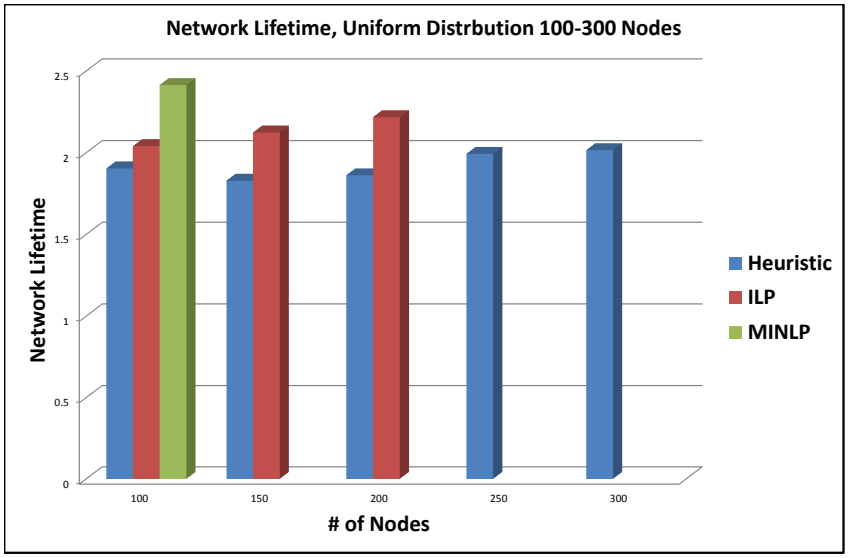

Figure 3.12: Network Lifetimes For Uniform Distribution

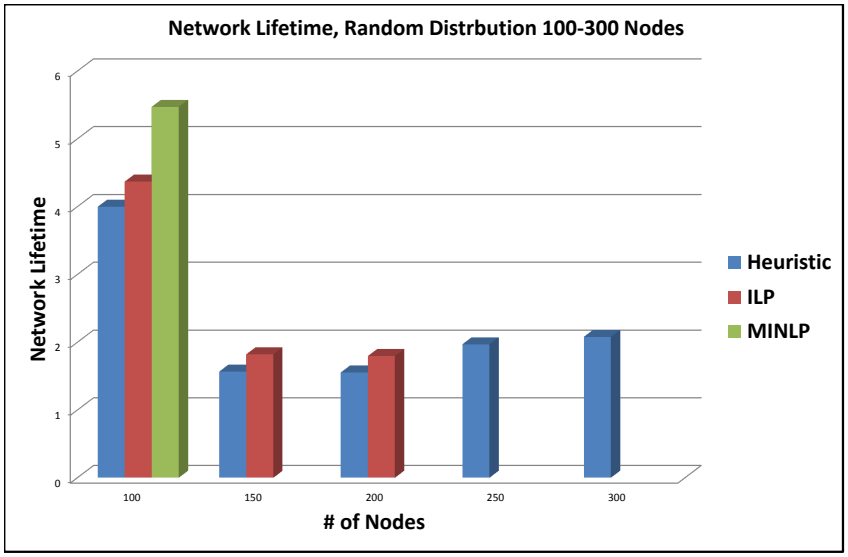

Figure 3.13: Network Lifetimes For Random Distribution

Figures 3.14, 3.15, and 3.16 show the running times of all the three approaches for all the instances with the three different distributions.

The simulation results show that the heuristic approach always constructs a shortest path overhearing tree in a reasonable amount of time for all the instances while both the MINLP-based approach and the ILP-based approach do not scale. For example, the ILP-based approach fails to construct a shortest path overhearing tree for the instances with more than 200 sensor nodes in 3 hours while the MINLP-based approach fails to construct a shortest path overhearing tree for the instances with more than 100 sensor nodes in 3 days.

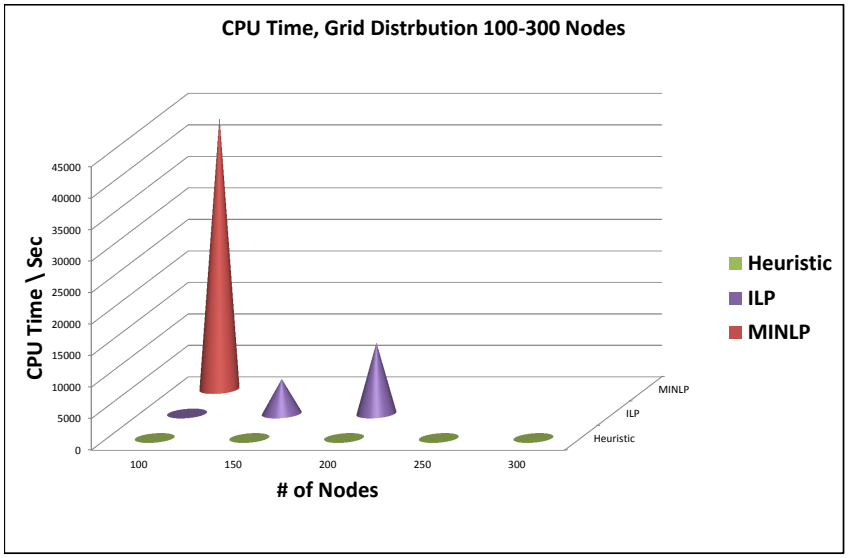

Figure 3.14: Running Times For Grid Distribution

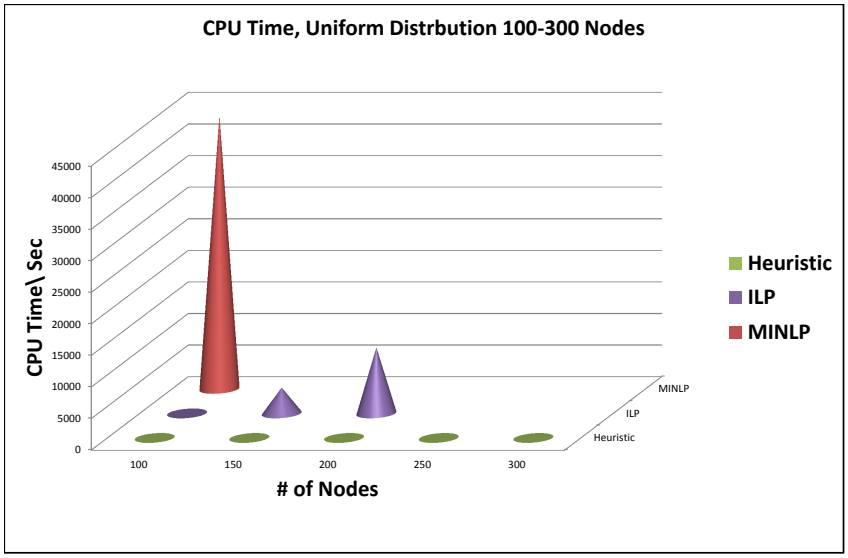

Figure 3.15: Running Times For Uniform Distribution

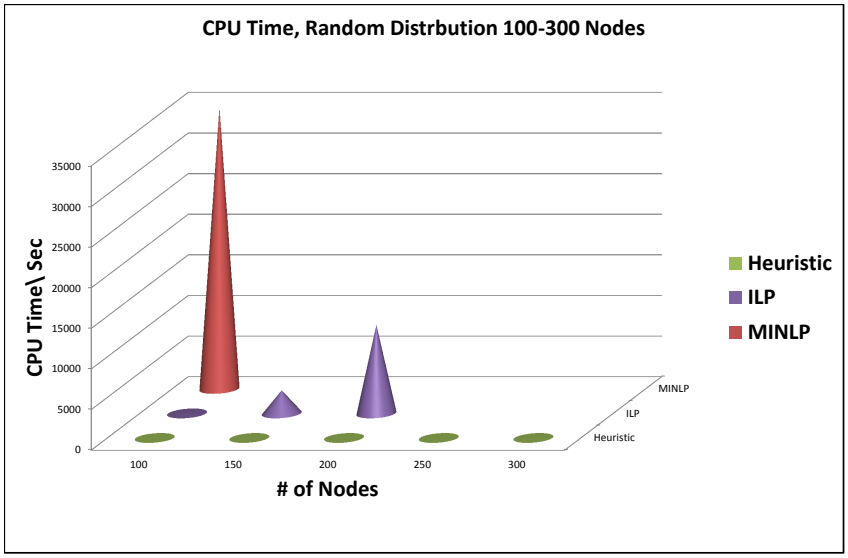

Figure 3.16: Running Times For Random Distribution

3.8 Chapter's Summary

We investigate the problem of constructing a shortest path overhearing tree with maximum network lifetime, and propose three approaches: a polynomial-time heuristic approach, a MINLP-based approach and an ILP-based approach. We have implemented our approaches using MIDACO solver and MATLAB Intlinprog, and performed extensive simulations on the 150 network instances with three different distributions: uniform, grid, and random distributions.

In homogeneous networks, the simulation results show that the average lifetime of all the network instances achieved by the heuristic approach is 85.69% of that achieved by the ILP-based approach and 81.05% of that obtained by the MINLP-based approach, and that the performance of the ILP-based approach is almost equivalent to that of the MINLP-based approach. In heterogeneous networks, the simulation results show that the average lifetime of all the network instances achieved by the heuristic approach is 87.371% of that achieved by the ILP-based approach and 74.9% of that obtained by the MINLP-based approach, and the performance of the ILP-based approach is almost equivalent to that of the MINLP-based approach.

We discuss the case when the monitor colludes with the parent of the monitored sensor node, where the attacks by the parent of the monitored sensor node may not be detected. Accordingly, we update our algorithms to select different monitors in different time intervals to allow the base station to investigate modification attacks and selective forwarding attacks.

Chapter 4

Reliable and Secure End-to-End Data Aggregation Using Secret Sharing in WSNs

Data aggregation in WSNs can effectively reduce communication overheads and reduce the energy consumption of sensor nodes. A WSN needs to be not only energy efficient, but also secure. Various attacks may make data aggregation unsecure. In this chapter, we investigate the reliable and secure end-to-end data aggregation problem considering selective forwarding attacks and modification attacks in homogeneous WSNs, and propose two data aggregation approaches. Our approaches, namely, Sign-Share and Sham-Share, use secret sharing and signatures to allow aggregators to aggregate the data without understanding the contents of messages and the base station to verify the aggregated data and retrieve the raw data from the aggregated data. To the best of our knowledge, this is the first lightweight en-routing malicious node detection in concealed data aggregation. We evaluate our approaches by comparing them with the two state-of-the-art approaches PIP and RCDA-HOMO.

4.1 Introduction

Data aggregation in WSNs (Wireless Sensor Networks) refers to the process of gathering data and representing it using a summary form. It can effectively reduce the data size, resulting in significant energy reduction in transmitting and receiving data.

Typically, a WSN is partitioned into clusters with a cluster head in each cluster [61]. Each cluster head gathers and aggregates the data received from its members, and sends the aggregated data to the base station.

There are many security requirements for data aggregation, including data confidentiality, data integrity, data freshness, data availability, authentication, and non-repudiation [62]. The contents of the data in transit should not be revealed to any party that is not authorized to have access [63]. Data confidentiality may be achieved via two different types of secure data aggregation schemes, namely, end-to-end scheme and hop-by-hop scheme. An end-to-end scheme does not use decryption when aggregating the data [64], and thus is more energy efficient. Several end-to-end data aggregation schemes have been proposed [65–69]. In a hop-by-hop scheme, a sensor node encrypts its data and sends the encrypted data to its aggregator. Each aggregator, after decryption, applies an aggregation function to aggregate the data, then encrypts it before sending it to another aggregator or the base station [70, 71]. Since encryption and decryption are computationally expensive, a hop-by-hop scheme may consume a significant amount of energy.

In WSNs, various attacks may exist. Among them are selective forwarding attacks and modification attacks. In the selective forwarding attacks, a malicious sensor node may deliberately drop some packets received from other sensor nodes, resulting in packet loss. In the modification attacks, a malicious sensor node may modify some packets received from other sensor nodes and forward the incorrect packets to the base station. Accordingly, the BS is required to identify the malicious packets as well as identifying the malicious aggregator.

Interestingly, most existing concealed data aggregations are not able to identify malicious aggregators. Some schemes exists to identify malicious aggregators [72–74]. However, those schemes establish a separated algorithm to identify malicious aggregator after discovering violation on the data aggregated. Unfortunately, establishing an identification algorithm for detection only waste sensors energy consumption, thus, decrees network lifetime. Beside that, it does not guarantee to discover the malicious aggregator as the malicious behavior is unpredictable and may perform well in the identification algorithm.

Malleability Resilient Concealed Data Aggregation in Wireless Sensor Networks [75] proposes an approach to discover the malicious aggregator within the routing scheme. In this approach, every sensor node provides its MAC (Message Authentication Code), and every receiver checks the validity of the MAC received from each packet. If the MAC is invalid, the previous aggregator performs malicious aggregation. Obviously, this approach is neither energy efficient nor delay efficient considering the fact that the more children exist, the more energy consumed for validation, and the more delay expected.

To the best of our knowledge, our approaches are the first to investigate the problem of identifying the malicious aggregator in concealed data aggregation in wireless sensor network by providing a lightweight secure concealed data aggregation with identification mechanism. This allows the BS to identify the malicious packet and malicious aggregator simultaneously. We make the following major contributions:

- We propose two secure data aggregation approaches for the end-to-end data aggregation in WSNs based on secret sharing and signatures. The proposed approaches can defend against both selective forwarding attacks and modification attacks.
- We propose an approach to identify the malicious aggregator in en-routing concealed data aggregation.
- We have compared both approaches and two state-of-the-art approaches, namely PIP [65] and RCDA-HOMO [67] using extensive simulations. In comparison, our ap-

proaches and the two state-of-the-art approaches PIP and RCDA-HOMO. The simulation results show both Sign-Share and Sham-Share consume reasonable amount of time in processing the data and aggregating the data. The simulation results show that our first approach achieves average network lifetime of 102.33% over PIP, and average aggregation energy consumption of 74.93%. Also, it achieves average aggregations processing time and sensor's data processing time of 95.4%, 90.34% over PIP and 98.7%,92.07% over RCDA-HOMO respectively while it achieved average network delay of 71.95% over PIP. Although RCDA-HOMO is completely a different technique, a comparison performed to measure the computational overhead.

4.1.1 Security Requirements in Data Aggregation

A secure WSN needs to satisfy the following security requirements:

- Data confidentiality: the contents of the information in transit should be revealed to any party that is not authorized to have access [63]. Data confidentiality may be divided into secure data aggregation schemes, end to end basis or hop-by-hop scheme. In the hop by hop scheme, data should be decrypted at every aggregator point [76]. Also, after decryption, an aggregation function is applied and the aggregation data then encrypted before sending it to the aggregator point. Extra computation in WSN makes this not feasible.

- Data integrity: message content should not be altered either maliciously or by accident during the transmission [77]. Data integrity builds on the confidentiality and assures the end user that the information received has not been altered while in transit and is as originally sent by the sender.

- Data freshness: A secure WSN needs to protect an aggregation scheme against replay attacks by ensuring that the data is recent and that no obsolete messages have been

received [62]. Data freshness builds on confidentiality and data integrity by ensuring that an adversity cannot replay a shared key.

- Data availability: A secure WSN needs to ensure that information in the network can be accessed when desired, and check whether the system is alive by eliminating compromised nodes.

- Authentication: both entity and data authentication should be supported in the model to support security. Entity authentication allows the message receiver to determine whether the said sender sends the message. Data authentication, on the other hand, guarantees that the data received is valid.

- Non-repudiation: it guarantees that once an aggregator sends its aggregation data, the aggregator cannot deny it. [78]. Therefore, the base station can determine the cause of any changes in the aggregation results.

- Data accuracy: the aggregated data should be provided as accurately as possible [61]. The worthiness of reducing the number of bits in the aggregator data is only realized if the data accuracy is high. During the design phase, data accuracy and aggregated data size trade-off should be considered since the need for high accuracy involves sending more bits, which results in more power consumption.

The rest of the chapter is organized as follows. Section 4.2 discusses the related work. Section 4.3 describes the network model, adversary model, and Boneh et al.s signature scheme. Section 4.4 proposes the first data aggregation approach Sign-Share. Section 4.5 presents the second secure data aggregation approach Sham-Share based on Shamir's secret sharing. Section 4.6 discusses the Sign-Share Malicious aggregator detection scheme while 4.7 discuss the security and scalability of our approaches. Section 4.8 shows the simulation results and analysis. Section 4.9 concludes this chapter.

4.2 Related Work

Three recoverable concealed data aggregation schemes, namely, RCDA-HOMO, RCDA-HETE, and Naive RCDA-HETE, are proposed in [67] to allow the base station to retrieve individual data of sensor nodes and provide data integrity and authenticity via signatures. These schemes combine the algorithm proposed by Mykletun et al. [79] with the algorithm proposed by Boneh et al. [80], and have four phases, namely, setup, encrypt-sign, aggregate, and verify while a naive scheme has five phases, namely setup, intracluster encrypt, intercluster encrypt, aggregate, and verify. Shim et al. [69] provide full analysis of the weaknesses of RCDA in both homogeneous and heterogeneous versions, and propose Sen-SDA for heterogeneous WSNs by changing the key generation schemes and related procedures.

DAA (Data Aggregation and Authentication) is proposed in [81] to provide secure data aggregation and false data detection in order to protect the network against an aggregator being compromised. The main idea behind DDA is to allow some nodes to monitor the aggregation operations and provide MAC (Message Authentication Code) for their results and include it as part of the MAC value of the cluster head, which implies that the final MAC value contains the MAC values for both the aggregator and the monitor. Moreover, each forwarding node is supposed to establish a pair of secret keys for the monitor node in order to verify the MAC value provided by its monitoring node.

An integrity protecting and hierarchical concealed data aggregation schema for WSNs has been proposed in [82]. The proposed algorithm, IPHCDA aggregates the aggregated data with different keys and goes through four procedures, namely key generation, encryption, aggregation, and decryption. Even though IPHCDA performs better than some other privacy homomorphic data aggregation schemes, but still has significant encryption and decryption overheads. Using the privacy homomorphism, a concealed data aggregation algorithm for reverse multicast traffic is proposed in [83]. The proposed algorithm intends

to provide integrity and end-to-end security as well as considering the risk of corrupted sensor nodes by proposing a key pre-distribution scheme. [84] proposes a simple secure and efficient aggregation algorithm for minimizing the network lifetime. The algorithm intends to reduce the size of the cipher by using the modular addition. [85] proposes two algorithms, namely, CPDA (Cluster-based Private Data Aggregation) and SMART (Slice-Mix-AggRegaTe), to bridge the gap between data privacy and collaborative data aggregation. Both algorithms intend to obtain a precise aggregation result.

Zhou et al. [66] propose a novel secure data aggregation mechanism which adopts a symmetric-key homomorphic encryption and combine it with homomorphic MAC to protect data and check the integrity of the aggregated data. [65] proposes a privacy and integrity preserving data aggregation scheme named PIP by combining Shamir's secret sharing and recursive secret sharing. PIP aims to provide privacy and integrity in data aggregation by hiding nodes details from others and neither encrypting the data nor using peer monitoring. PIP intends to prevent an aggregator from understanding the contents, even after receiving all shares, by scrambling the shares. It generates three different keys, perturbation key, integrity key, and scramble key. [86] presents three mechanisms based on multipath routing where the first scheme guarantees confidentiality of the data by using straightforward secret sharing, and the other two schemes provide data availability by using information dispersal.

Different protocols and schemes have been proposed in the field of secure data aggregation using concealed data aggregation. First of all, EIRDA [87] which used a reputation exchange to create a trusted network such that a trusted path is found towards the aggregator within a cluster. However, EIRDA introduces a significant loss of energy in presence of some denial of service attacks [88].

All the previous schemes cannot handle both the selective forwarding attacks and the modification attacks in homogeneous cluster-based WSNs.

Renewable hash chain proposed in [89] to provide a security data aggregation scheme in hierarchical WSNs. Data authentication to provide both confidentiality and integrity is taken into consideration. Although the proposed scheme dose not define well against modification attacks and selective forwarding attack, but it still provides security against replay attacks and tempering attacks as it authenticate the data within the cluster and between clusters which provide some length of security and significant waste of energy.

Malleability resilient concealed data aggregation protocol has been proposed in [75] using concealed data aggregation to defined against active and passive attacks. The proposed scheme used homomorphic encryption and homomorphic message authentication code. This protocol required that every intermediate node verify the data of the previous layer before it aggregate it and forward it. One of the major limitation is that each aggregator is required to verify the integrity, perform aggregation, and forwarding which may cause a significant waste of energy especially when the number of children increases. On the other hand, the use of one aggregator is not sufficient, as compromising leaf node and lower level aggregator completely destroy the scheme. FESA [90] also uses the message authentication code (MAC)integrity. Moreover, FESA also uses MFN-group such that the network is divided into groups which consist of monitoring, forwarding, and neighboring node to investigate whether the data has been modified. However, FESA has a high communication cost and no data loss resiliency [88]. Similarly, SA-SPKC [91] used symmetric key-based homomorphic encryption with a huge packet size [88] which consumes extra energy consumption.

Secure In-network processing of Exact Sum (SIES) queries has been proposed in [72] using homomorphic encryption and secret sharing to provide integrity and confidentiality of the aggregated data. SIES also meant for malicious node identification such that when the BS detect the data violation, it establish a separated algorithm to identify the malicious node. However, establishing an algorithm for identification only is a total waste of energy, and it causes delay and loss of data freshness. Moreover, it does not guarantee the identification

as the malicious node may challenge the network during the process of the identification algorithm.

4.3 Network Model and Attack Model

The target WSN consists of a set $V = \{v_1, v_2, \cdots .v_n\}$ of n static sensors. There is only one base station. The network is divided into disjoint clusters [92–94]. We assume that two aggregators are selected among all the members for each cluster. Each aggregator sends its aggregated data to the base station directly.

Sensor nodes may be compromised. A compromised sensor node is called a malicious sensor node. A malicious sensor node may drop, or modify the packets it receives from other sensor nodes.

We use the following attack model:

1. Without compromising either a sensor node or an aggregator, an adversary can eavesdrop the data in transmission.
2. If an aggregator is compromised, it may modify the data received from a child node, or drop a packet selectively.
3. If a sensor node is compromised, the adversary can obtain its keys and thus calculate the keys of other sensor nodes.

4.3.1 Boneh et al.'S Signature Scheme

A signature scheme is introduced in [80] and used in different approaches [67, 68]. The scheme consists of 5 phases, starting with the key generation, followed by signature, veri-

fying, aggregation, and finally verifying aggregated signature. Details of the algorithm are given in Algorithm.1.

Algorithm 1 Boneh et al. Procedure for entity i

- KeyGen(τ): τ is a security parameter.

1: Generate Private Key pr_i randomly selected from $\boldsymbol{Z_p}$.

2: Generate Public key $pu_i \in G_2$ where $pu_i = pr_i$ x g_2

3: Output key pair (pr_i, pu_i) for entity i

- Verification: Given user's public key pu, a message M, and a signature σ, compute $h \leftarrow h(M)$; accept if $e(g_1, \sigma) = e(v, h)$ holds.

- Sign$_{x_i}$ Sign message m with the private key pr_i.

1: Compute $h = H(m)$, where $h \in G_1$.

2: Generate Signature $\sigma = pr_i$ x h and return (m, σ).

- Agg$_k(\delta, M)$: Aggregate k signatures where message set $M = \{m_1, ..., m_k\}$; m_i for entity i and $\delta = \{\sigma_1, ..., \sigma_k\}$; σ_i is signature of m_i.

1: Produce aggregated signature $\hat{\sigma} = \sigma_1 + ... + \sigma_k = \sum_{i=1}^{k} \sigma_i$, where $\hat{\sigma}$, $\sigma_1, ..., \sigma_k \in G_1$

- Agg-Verify$_\vartheta(\hat{\sigma}, M)$: Verify the aggregated signature.

$\hat{\sigma}$ is the aggregated signature of message set M where $M = \{m_1, ..., m_k\}$; m_i from entity i, and public key set $\vartheta = \{pu_1, ..., pu_k\}$; $pu_i \in U_i$.

1: Compute $h_i = H(m_i)$, for $1 \leq i \leq k$.

2: Accept if $e(\hat{\sigma}, g_2) = \prod_{i=1}^{k} e(h_i, pu_i)$, where $e(\hat{\sigma}, g_2)$, $e(h_i, pu_i) \in G_T$; Otherwise, Reject.

4.4 Sign-Share

In our Sign-Share approach, each sensor node splits its data into multiple shares and sends them to the aggregators of its cluster, allowing encoding each share with simpler codes.

For ease of description, we assume that the data sensed by each sensor, each time is 32-

bits long, and the 32-bits data is split into four 8-bit shares. However, a trade-off found between security and energy efficiency such that the more shares generated, the lengthen security, and shorten the network lifetime.

Our Sign-Share approach consists of the following phases:

Setup Phase: The following system parameters are generated and loaded into each sensor node at the design stage.

- A secret key-set K in a form of matrix shown as follows:

$$K = \begin{bmatrix} \lambda_0 & \mu_0 \\ \lambda_1 & \mu_1 \\ \lambda_2 & \mu_2 \\ \lambda_3 & \mu_3 \end{bmatrix} 0 \leq \lambda_k, \mu_k < P$$

  The larger the P, the more secure the aggregations.

- A secret 32-bit pseudo random binary sequence generator $PRBS_p[I, n]$, where I is the seed and n is the clock.

- (pu_{v_i}, pr_{v_i}): this pair is generated according to the algorithm proposed by Boneh et al. [80]. However, the private key pr_{v_i} is set to λ_0.
  - pu_{v_i}: the public key which is kept at the base station.
  - pr_{v_i}: the private key which is loaded to each sensor node v_i.

- A hash function H for all the sensor nodes.

Secret Sharing-Signature Phase: When a senor node v_i senses the physical environment and prepares its data D to be sent to its aggregators, it does the following:

- Each sensor v_i splits its data as follows:

1. Encode the data: $D' = D \oplus PRBS_p[I, n]$, where $\oplus$ is the bitwise XOR.
2. Split the encoded data into 4 shares B_0, B_1, B_2, and B_3.
3. Encode each share B_k using the key-set K as follows:

$$B'_k = ((B_k * \lambda_k) + \mu_k) \mod 256 \tag{4.1}$$

- Sign each share as follows:

$$h_i = H(B'_k) \tag{4.2}$$

$$\sigma_i = pr_{v_i} * h_i \tag{4.3}$$

- Send the data in a tuple (B'_k, σ_i) to each aggregator of its cluster such that the data after encoding is split equally between them.

Aggregation Phase: When an aggregator node receives the tuple from every member of its cluster, it does the following:

- Let $(B'_0, \sigma_0), (B'_1, \sigma_1), \cdots, (B'_{w-1}, \sigma_{w-1})$ be all the tuples received.
- Aggregate the signatures as follows:

$$\hat{\sigma} = \sum_{i=1}^{w} \sigma_i \tag{4.4}$$

- Aggregate all the shares as follows:
  - Concatenate the w shares into a single value Q as follows:

$$Q = B'_0 | B'_1 | ... | B'_{w-1} \tag{4.5}$$

- Send the concatenated data in a tuple $(Q, \hat{\sigma})$ to the base station.

Verification-Decoding Phase: When the base station receives the data from every aggregator AG_i, it does the following:

- Let w be the number of shares received from AG_i.
- Extract the Q bytes of each tuple received from AG_i.
- Recover the 32-bit data of each node v_i as follows:
  1. Decode each byte using the key-set K of v_i:

$$B_k = ((B'_k - \mu_k) * \lambda_k^{-1}) \mod 256 \tag{4.6}$$

  2. Merge the decoded bytes into one 32-bit integer D'.
  3. Decipher the data: $D = D' \oplus PRBS_p[I, n]$.
- Verify D by using Boneh et al. algorithm [80].

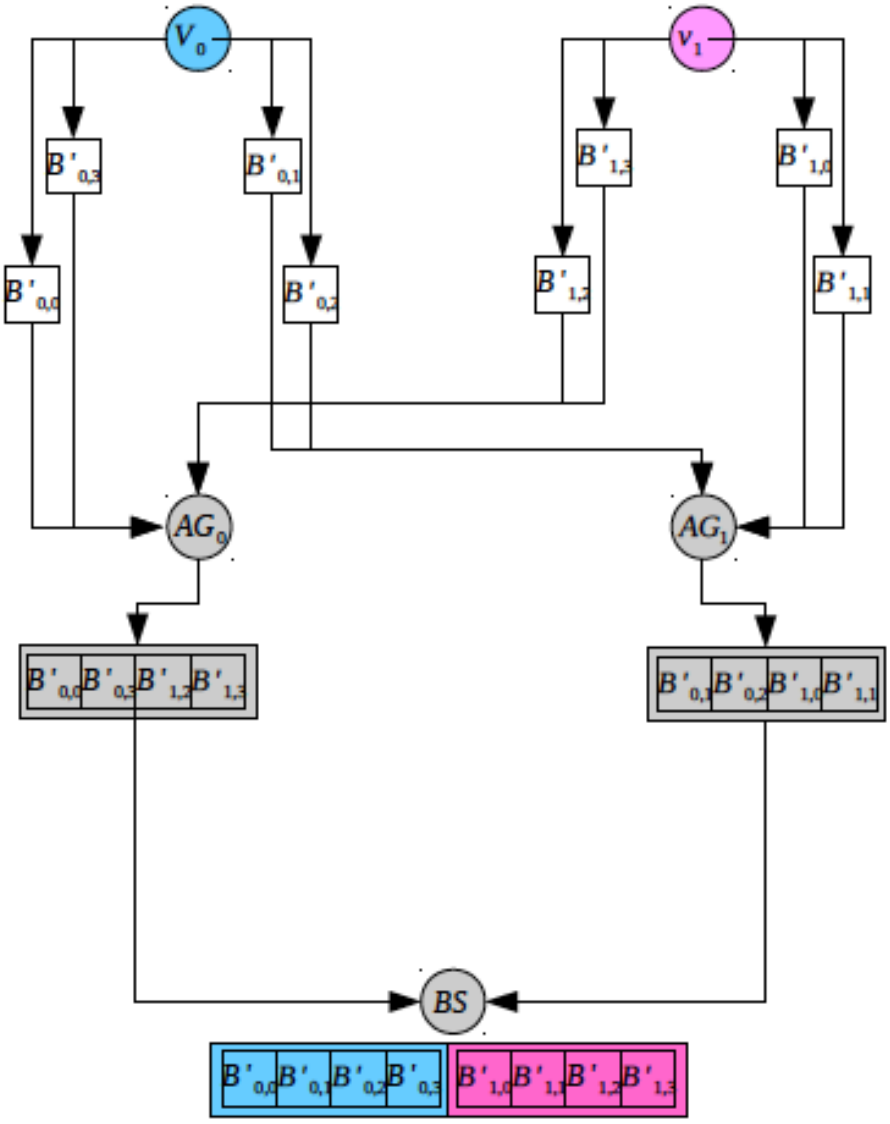

Figure 4.1: Example Of The Scheme

4.4.1 A Numerical Example

Consider Figure 4.2, where the sensed data by sensor node v_1 is 127. The Key-set K of sensor node v_i is loaded and shown in the right hand side along with the PRBS value. The data is split into four shares and encoded as explained before.

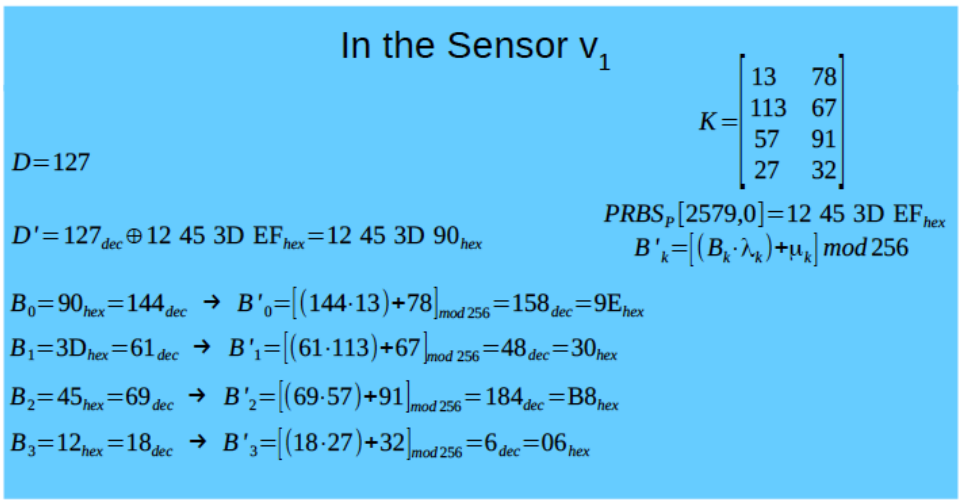

Figure 4.2: Packet Preparation In a Sensor Node

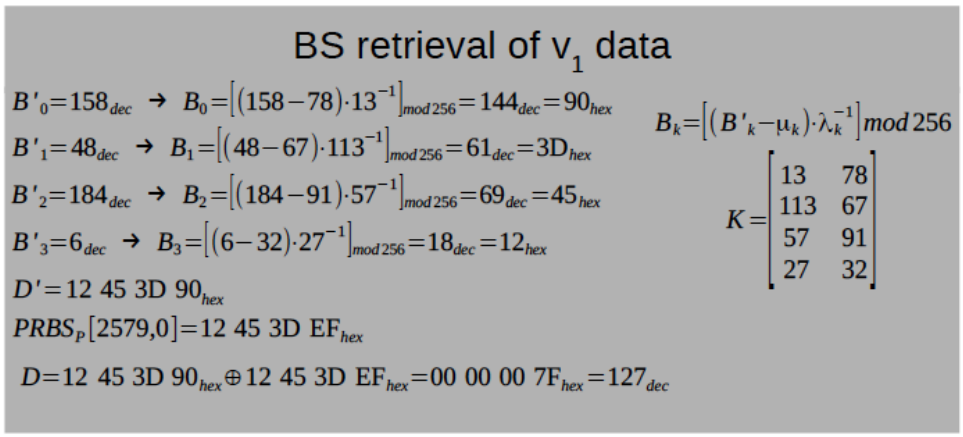

Figure 4.3: Base Station Verification

4.5 Sham-Share

Our Sham-Share approach consists of the following phases:

Setup Phase: The base station generates the following key pair (pu_{v_i}, pr_{v_i}) for each sensor node v_i as in [80], where pu_{v_i} is the public key kept in the base station, and pr_{v_i} is the private key loaded to each sensor node v_i along with H, the hash function for all the

sensor nodes.

Secret Sharing-Signature Phase: When a senor node v_i senses the physical environment and prepares its data S to be sent to its aggregators, it performs the following tasks:

- The sensor node v_i splits the data S into 4 shares as follows:
  1. Generate two random numbers a_0, a_1.
  2. Construct the following polynomial function:

$$f(x) = S + a_0 x + a_1 x^2 \tag{4.7}$$

  3. Construct 4 shares with each share represented by a pair $(x, f(x))(x = 1, 2, 3, 4)$. Shares start from $(1, f(1))$ because $f(0)$ is the data S.
  4. Let ID_i be the ID of the sensor node v_i. Encode each share of v_i as follows:

$$Q_i = x + 10id_i + 1000f(x) \tag{4.8}$$

- Sign each share as follows:

$$h_i = H(Q_i) \tag{4.9}$$

$$\sigma_i = pr_{v_i} * h_i \tag{4.10}$$

- Send the tuples (Q_1, σ_1), (Q_2, σ_2) to one aggregator, and (Q_3, σ_3), (Q_4, σ_4) to the other aggregator.

Aggregation Phase: After an aggregator AG_i receives the tuple from every member of its cluster, it performs the following tasks:

- The aggregator gathers all the w tuples (Q_0, σ_0), (Q_1, σ_1), ..., (Q_{w-1}, σ_{w-1}) from the members of its cluster.

- Aggregate the signatures as follows:

$$\hat{\sigma} = \sum_{i=1}^{w} \sigma_i \tag{4.11}$$

- Send the data in an array which contains the aggregated signature and the aggregated shares.

$$\begin{bmatrix} \hat{\sigma} \\ Q_0 \\ Q_1 \\ ... \\ Q_{w-1} \end{bmatrix}$$

Reconstruction-Verification Phase: After the base station receives the data from all the aggregators, it performs the following tasks for each aggregator AG_i:

- Let w be the number of shares received from AG_i.
- Disaggregate Q_i of each array received from AG_i as follows:

$$\begin{bmatrix} f(x_0), id_0, x_0 \\ f(x_1), id_1, x_1 \\ ... \\ f(x_{w-1}), id_k, x_{w-1} \end{bmatrix} =$$

$$\begin{bmatrix} \lfloor Q_0/1000 \rfloor, \lfloor Q_0/10 \rfloor \mod 100, Q_0 \mod 10 \\ \lfloor Q_1/1000 \rfloor, \lfloor Q_1/10 \rfloor \mod 100, Q_1 \mod 10 \\ ... \\ \lfloor Q_{w-1}/1000 \rfloor, \lfloor Q_{w-1}/10 \rfloor \mod 100, Q_{w-1} \mod 10 \end{bmatrix}$$

- Gather 3 shares of each sensor node v_i, and reconstruct its data S as follows:

$$S = \sum_{j=0}^{2} f(x_j) \prod_{m=0, m\neq j}^{2} \frac{x_m}{x_m - x_j} \tag{4.12}$$

- Verify S by using Boneh et al. algorithm [80].

4.5.1 A Comprehensive Example

For simplicity, we consider a scenario of 3 sensor nodes v_0, v_1 , and v_2 and 2 aggregators AG_1, andAG_2. Each sensor node senses the physical environment to obtain data denoted by S. In this example, the sensed data S forv_0, v_1 , and v_2 are 4,7, and 9, respectively. Accordingly, each node creates its polynomial function and generates a random number a_0, followed by creating shares, and then uses the decimal concatenation as follows:

- $f(x_{v_{00}}) = 4 + 2x \rightarrow D_0(1, 6), D_1(2, 8)$
  - Using decimal concatenation

$$\rightarrow v_{00} = (Q_{0,0}(6001), Q_{0,1}(8002)) \tag{4.13}$$

- $f(x_{v_{01}}) = 7 + 3x \rightarrow D_0(1, 10), D_1(2, 13)$
  - Using decimal concatenation

$$\rightarrow v_{01} = (Q_{1,0}(10011), Q_{1,1}(13012)) \tag{4.14}$$

- $f(x_{v_{02}}) = 9 + 7x \rightarrow D_0(1, 16), D_1(2, 23)$
  - Using decimal concatenation

$$\rightarrow v_2 = (Q_{2,0}(16021), Q_{2,1}(23022)) \tag{4.15}$$

Then a sensor node signs one share to each aggregator in pairs $\{(Q_{0,0}, \sigma_{0,0}) ....(Q_{2,1}, \sigma_{2,1})\}$.

Next, each sensor node sends its data to the two aggregators where each aggregator is supposed to produce an array of the aggregated result such that, the following arrays are sent to the BS from AG_1 and AG_2:

$$\begin{bmatrix} \hat{\sigma}_i \\ 6001 \\ 10011 \\ 16021 \end{bmatrix} \begin{bmatrix} \hat{\sigma}_i \\ 8002 \\ 13012 \\ 23022 \end{bmatrix}$$

After the base station receives both packets, it de-aggregates the signature and verifies the data after reconstructing the original data. Accordingly, the de-concatenation step is performed to bring the shares back to their original forms and allow the base station to reconstruct the original data and compare it with the signed data. According to matrix $[M]$, the first value computes $f(x)$, the second value calculates the node ID, and the third value figures out the value of x.

$$AG_1 = \begin{bmatrix} 6 & 00 & 1 \\ 10 & 01 & 1 \\ 16 & 02 & 1 \end{bmatrix} AG_2 = \begin{bmatrix} 8 & 00 & 2 \\ 13 & 01 & 2 \\ 23 & 02 & 2 \end{bmatrix}$$

- $v_{00} = D_0(1, 6), D_1(2, 8)$
- $v_{01} = D_0(1, 10), D_1(2, 13)$
- $v_{02} = D_0(1, 16), D_1(2, 23)$

Now, the base station is ready to retrieve and reconstruct the data and compare it to the hashed value in the signature.

- $S(v_{00}) = 6 * \frac{2}{2-1} + 8 * \frac{1}{1-2} = 4$
- $S(v_{01}) = 10 * \frac{2}{2-1} + 13 * \frac{1}{1-2} = 7$
- $S(v_{02}) = 16 * \frac{2}{2-1} + 23 * \frac{1}{1-2} = 9$

4.6 Sign-Share Malicious Aggregator Detection

In this section we modified the aforementioned Sign-Share algorithm to allow the BS to detect the malicious aggregator which intend to modify or selectively drop the data. We open up the assumption from direct transmission from aggregators to the BS to multi-hop routing which involved multi-aggregator in a single path.

To this end, we employ homomorphic message authentication code (MAC) [95] which helps to verify the integrity of aggregated data in our concealed data aggregation scheme.

Setup Phase: In addition to the assumptions presented in Section 4.4, we assume that each sensor node i shares a pair-wise symmetric key $k_{i,j}$ with its cluster head.

Secret Sharing-Signature Phase: When a senor node v_i senses the physical environment and prepares its data D to be sent to its aggregators, it does the following:

- Each sensor v_i splits its data as follows:

  1. Encode the data: $D' = D \oplus PRBS_p[I, n]$, where $\oplus$ is the bitwise XOR.
  2. Split the encoded data into 4 shares B_0, B_1, B_2, and B_3.
  3. Encode each share B_k using the key-set K as follows:

$$B'_k = ((B_k * \lambda_k) + \mu_k) \mod 256 \tag{4.16}$$

- Node v_i computes the following homomorphic MAC tag for each byte:

$$t_i = \text{HomMAC}(B'_k)$$

  where the HomMAC function generates a homomorphic MAC based on an algorithm presented in [95].

- Node v_i encrypts the homomorphic MAC tag t_i with a symmetric algorithm, such as AES, to obtain $\text{E}_{k_{i,j}}(t_i)$, where node v_j is the cluster node for node v_i.

- Node v_i sends the data in a tuple $(B'_k, \mathrm{E}_{k_{i,j}}(t_i))$ to the aggregator nodes, such as v_j, in its cluster such that the data after encoding is split equally between several aggregator nodes.

Aggregation Phase: An aggregator node v_j receives the tuple $(B'_i, \mathrm{E}_{k_{i,j}}(t_i))$ from each child node v_i. Let us assume that w denotes the number of child nodes for the aggregator node v_j. Now the aggregator node runs the following steps:

- The aggregator computes the homomorphic MAC tag over each data B'_i received from the child node v_i, and compares the computed MAC tag to the decrypted MAC tag received in the tuple.

$$\mathrm{D}_{k_{i,j}}(\mathrm{E}_{k_{i,j}}(t_i)) = \mathrm{HomMAC}(B'_i)$$

- If all the received MAC tags from child nodes are valid, the aggregator node v_j computes the aggregated data and MAC tag as follows:

$$B'_j = \bigoplus_{w}^{i=1} B'_i$$

$$t_j = \bigoplus_{w}^{i=1} \mathrm{D}_{k_{i,j}}(\mathrm{E}_{k_{i,j}}(t_i))$$

- The aggregator node v_j encrypts the homomorphic MAC tag t_j with a symmetric algorithm, such as AES, to obtain $\mathrm{E}_{k_{l,j}}(t_j)$, where node v_l is the cluster node for node v_j.

- The aggregator node v_j sends the data in a tuple $(B'_j, \mathrm{E}_{k_{l,j}}(t_j))$ to the aggregator nodes, such as v_l, in its cluster such that the data after encoding is split equally between several aggregator nodes.

Base Station Phase: When the base station receives the data from every aggregator v_i, it performs similar steps described in Aggregation Phase to verify the MAC tag for each received data. If all the received MAC tags are valid, the base station obtains the aggregated value using a similar method presented in Section 4.4.

4.7 Security And Scalability Analysis

In this section, we show that our approaches are secure under the adversary model shown before.

Firstly, without compromising either a sensor node or an aggregator, an adversary is unable to understand the content of a message in transit between the source and destination since the data is encoded at each sensor node and split into shares.

Secondly, if an aggregator is compromised, it still cannot understand the content of the whole data as each sensor node only sends a subset of shares to each aggregator. The only case where the whole data can be deciphered is that both aggregators are compromised and collaborate to decipher the whole data.

Next we show how our approaches can defend against the proposed attacks.

- Scenario 1: one aggregator is compromised and modifies the data.
  - Both approaches can detect the modification attack by validating the signatures. However, Sham-Share gives extra strength to the base station. If the aggregator modifies some shares and the number of correct shares the base station receives is at least 3, the base station is able to not only detect the attack but also reconstruct the original data.
- Scenario 2: one aggregator is compromised and selectively drops some data.
  - The base station knows the number of sensor nodes in each cluster. Therefore, the base station expects a fixed number of shared from the two aggregators of each cluster. If an aggregator drops any shares, the number of shared received by the base station does not match with the number of shared expected. As a result, the base station will detect the modification.
- Scenario 3: both aggregators are compromised.

- If both aggregators drop some data, the base station will detect it as the base station expects to receive a fixed number of shares for each clusters.
- If one aggregator drops some data and the other aggregators modifies the data, the base station will detect which aggregator modifies the data and which aggregator drops the data. There are two reasons. Firstly, the base station expects to receive a fixed number of shares from each aggregator. Secondly, the base station can detect the modification by validating the signatures.

4.7.1 Probability for Guessing Keys

An attacker must successfully guess the keys stored in the sensor nodes in order to understand the original data. For example, the Sign-Share algorithm protects the message by making the keys invisible to the attackers.

- In order for an attacker to understand the content of a share, the attacker must guess both μ_k and λ_k correctly. The probability of guess both μ_k and λ_k correctly for one share is as follows:

$$P(\mu_k) * P(\lambda_k) = \frac{1}{p} * \frac{1}{p} \tag{4.17}$$

- Fortunately, even if the attack can guess this, understanding the content of the original data requires guessing the keys for both μ_k and λ_k for all the four shares. The probability is calculated as follows:

$$(P(\mu_k) * P(\lambda_k))^4 = (\frac{1}{p} * \frac{1}{p})^4 = p^{-8} \tag{4.18}$$

- Even after successfully guessing and understanding the content of all the four shares, the attacker cannot recover the original data, because the data is encoded with the PRBS value through the XOR function which is a 32-bit binary random sequence before dividing them into shares. After guessing all the aforementioned keys for the

generated shares, attacker must put the outcome value of all shares together, then guesses the PRBS value. The probability of guessing the PRBS value is computed as follows:

$$P(PRBS) = \frac{1}{2^{32}} \tag{4.19}$$

Consequently, an attacker needs to guess all the segments of μ and λ keys as well as the PRBS value. Thus, the probability of guessing these values is $2^{-32} \times p^{-8}$.

4.8 Simulation Results

4.8.1 Setup

In order to evaluate our approaches, we use the following performance metrics, the data processing time of a sensor node, the aggregation processing time of an aggregator, the energy consumption on data processing of a sensor node, the energy consumption on data aggregation of an aggregator, the network delay, and the network lifetime. The processing time of a sensor node is the time for a sensor node to prepare its packet based on the proposed scheme. The aggregation processing time is the time for an aggregator to aggregate the received packets based on the proposed scheme. The average energy consumption on sensor data processing is the average energy consumption on processing and sending the data. The average energy consumption on data aggregation is the average energy consumption on receiving, aggregating and sending data. The network delay is the time for the sensed data of a sensor node to travel to the base station. The network lifetime is the time until the first sensor node depletes its energy [96].

We use NS 3.22 to generate 6 instances of WSNs with uniform distribution. We vary the number of sensor nodes from 50 to 300 with an increment of 50. For each instance, sensor nodes are deployed in a 1000x1000 m^2 rectangular area, and the base station is deployed

at the center of the rectangular area. For each instance, 1000 transmissions are recorded. We use two aggregators per cluster and 4 shares for the data sensed by each sensor node. The energy consumption on sending one bit of data is $TX = 0.6\mu J$ while the energy consumption on receiving one bit of data is $RX = 0.67\mu J$. The energy consumption on processing one bit of data is $\omega = 0.47\mu J$ while the initial energy for every sensor node is $E_i = 100J$.

The hardware platform is Intel Core i5-3470 with a clock frequency of 3.20 Ghz, a memory size of 8 GB and a cache size of 8134 MB while the processor used in each sensor node is the default processor provided by NS3.22 with a clock rate of 1.2 GHz.

4.8.2 Results and Analysis

In order to evaluate our approach, we compare it with the state-of-the-art approaches RCDA-HOMO proposed in [67] and PIP proposed in [65] for the following two reasons. Firstly, PIP also uses Shamir's secret sharing algorithm. Secondly, PIP and RCDA are the most relevant and recent works on secure data aggregation. In this section, we show the comparison results of our approaches, RCDA-HOMO and PIP.

Figure 4.4 shows the average processing time while Figure 4.5 shows the energy on data processing for all the approaches. Sign-Share and Sham-Share have less processing time and less energy consumption on data processing than both PIP and RCDA-HOMO.

The average processing times of sensor nodes for Sign-Share, Sham-Sign, PIP and RCDA-HOMO are 2888.98 ms, 3040.84 ms, 3198.1 ms, and 3137.85 ms, respectively, while the average energy consumption is 1357.8 μJ, 1429.1 μJ,1503 μJ, 1474.8 μJ, respectively.

The minimum ratio between the sensors data processing time achieved by Sign-Share and the sensors data processing time obtained by PIP is 89.45%, and the maximum ratio is 91.3%. The average ratio for all the instances is 90.4%. On the other hand, the minimum,

the maximum, and the average ratio between Sham-Share and PIP are 94.5%, 95.65%, and 95%, respectively.

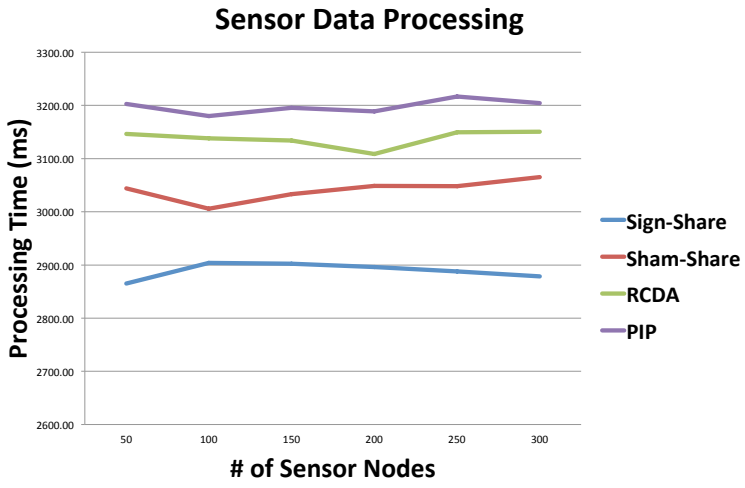

Figure 4.4: Data Processing Time for Sensor Nodes

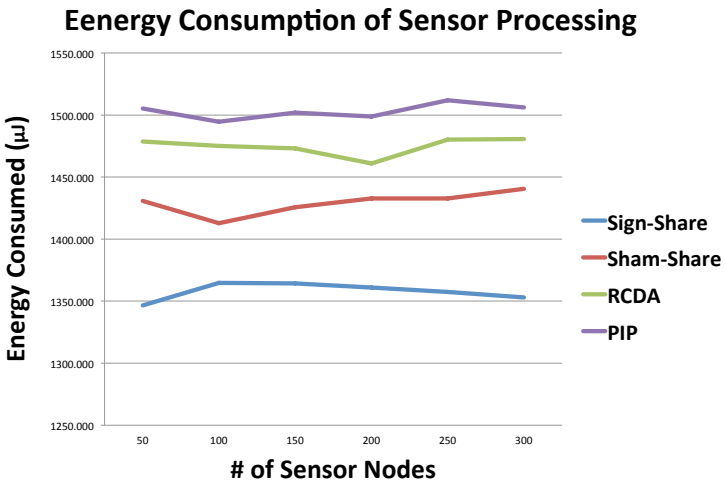

Figure 4.5: Energy Consumption for Sensor Processing

Figure 4.6 shows the average aggregation processing time while Figure 4.7 shows the energy consumption on data aggregation. The average processing times of Sign-Share, Sham-Share, RCDA-HOMO and PIP are 612.97 ms, 624.987 ms, 619.11 ms and 636.863 ms, respectively, while the average total energy consumption is 1243.3 μJ, 1449.9 μJ, 1045.2 μJ, 1656.4 μJ, respectively.

In terms of the total energy consumption, RCDA-HOMO consumes less energy than the other three approaches as it does not split the data into multiple shares. Moreover, our approaches still consume less aggregation energy than PIP.

In comparison between the Sign-Share approach and the PIP approach, the minimum ratio and the maximum ratio between the aggregator's data processing time achieved by the Sign-Share approach and the aggregator's data processing time obtained by the PIP approach are 90.91% and 97.61%, respectively. The average ratio for all the instances is

95.387%. On the other hand, the minimum, the maximum, and the average ratios between Sham-Share and PIP are 95.72%, 98.79%, and 97.73%, respectively.

In comparison between the Sign-Share approach and the RCDA-HOMO approach, the minimum ratio and the maximum ratio between the aggregator's data processing time achieved by the Sign-Share approach and the aggregator's data processing obtained by the RCDA-HOMO approach are 97.378% and 99.38%, respectively. The average ratio for all the instances is 98.74%. On the other hand, RCDA-HOMO performs slightly better than the Sham-Share approach.

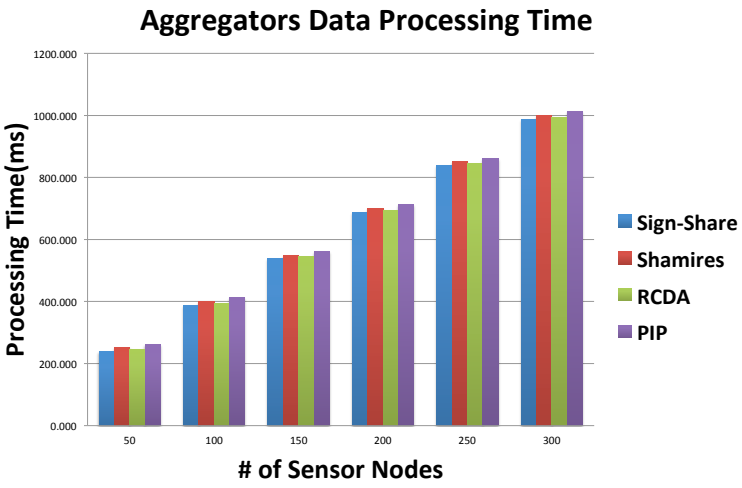

Figure 4.6: Aggregation Processing Time

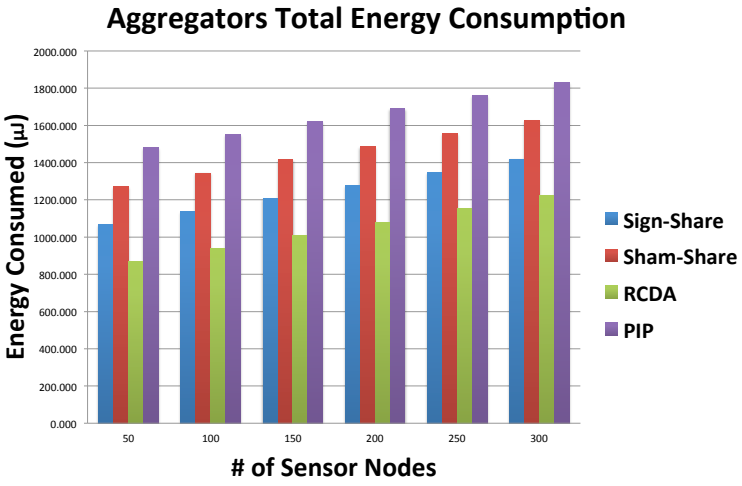

Figure 4.7: Aggregator's Energy Consumption

Figure 4.8 compares the network lifetime while Figure 4.9 shows the network delay among all the approaches. Again, since RCDA-HOMO consumes less energy on aggregators, it has slightly longer network lifetime than Sign-share and Sham-Share while PIP has the smallest network lifetime.

The average network lifetimes by the Sign-Share approach, the Sham-Sign approach, PIP and RCDA-HOMO are 108.5 time units, 109.67 time units, 106 time units, and 115.2 time units, respectively.

In comparison between the Sign-Share approach and the PIP approach, the minimum ratio, the maximum ratio, and the average ratio between the network lifetime achieved by the Sign-Share approach and the network lifetime obtained by the PIP approach are 92.56%, 100% and 97.85%, respectively. On the other hand, the minimum ratio, the maximum ratio, and the average ratio between Sham-Share and PIP are 92.1%, 99.05%,

and 96.7%, respectively.

The average network delays by the Sign-Share approach, the Sham-Sign approach, PIP, and RCDA-HOMO are 1509.2 ms, 1479.92 ms, 2182.95 ms, and 1275.92 ms, respectively.

In comparison between the Sign-Share approach and the PIP approach, the minimum ratio, the maximum ratio and the average ratio between the network delay achieved by the Sign-Share approach and the network delay obtained by the PIP approach are 59.65%, 88.399% are 71.95%, respectively. On the other hand, the minimum, the maximum, and the average ratio between Sham-Share and PIP are 60.19%, 90.32%, and 70.696%, respectively.

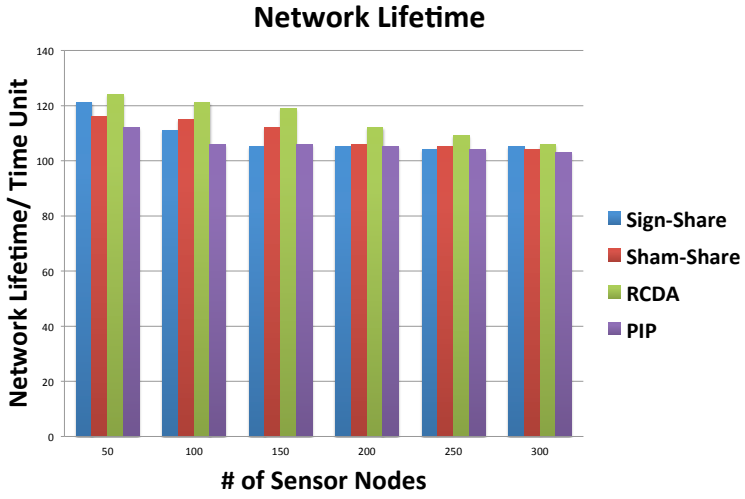

Figure 4.8: Network Lifetime

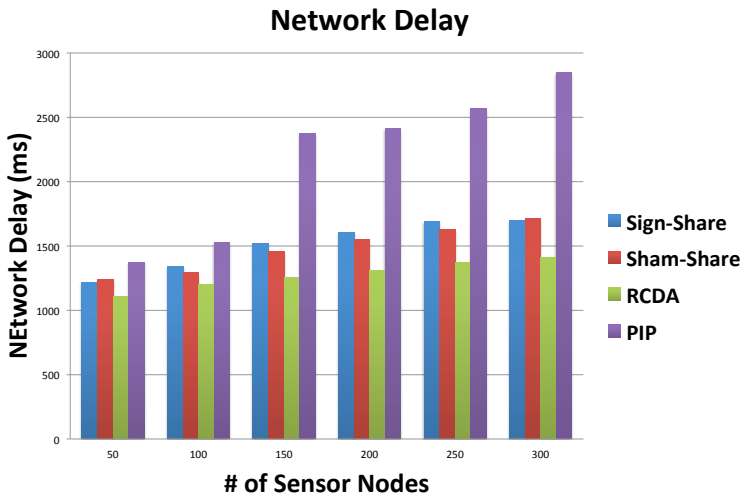

Figure 4.9: Network Delay

4.9 Chapters Summary

We investigate the problem of reliable and secure end-to-end data aggregation, considering selective forwarding attacks and modification attacks in homogeneous WSNs, and propose two data aggregation approaches that not only conceal the sensed data but also allow the base station to detect both the selective forwarding attacks and the modification attacks. We also modify the Sign-Share approach to detect the malicious aggregator in a multi-hop routing where the packet travels through different aggregators before reaching the base station.

The simulation results shows that both of our approaches perform better than PIP and RCDA-HOMO in terms of the aggregation processing time and the sensor processing time, and they perform significantly better than PIP in terms of the network lifetime, the

network delay, and the aggregation energy consumption.

One limitation with our approaches is that aggregators consume much more energy than other sensor nodes. As a result, they will die much sooner. In order to increase the network lifetime, we need to rotate aggregators. Another limitation with our approaches is that aggregators send the aggregated data to the base station directly. There are two major problems with direct communication between aggregators and the base station. Firstly, aggregators consume a large amount of energy due to long distance communication, especially in a large WSN. Secondly, sensor nodes typically have a limited communication range in order to save energy, and the aggregators far from the base station may not communicate with the base station directly. Therefore, a routing topology such as tree is desirable.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this thesis, we investigate three different problems related to secure data collection in WSNs under two attacks, the selective forwarding attack and the modification attacks. The first problem is to minimize the failure rate packet delivery in presence of the aforementioned attacks. The second problem is to construct a shortest path overhearing topology with maximum network lifetime. An the third problem is the secure end-to-end data aggregation considering the selective forwarding attacks and the modification attacks in homogeneous cluster-based WSNs. There are two major goals. The first goal is to protect the data from being modified or dropped, and the second goal is to maximize the network lifetime.

In Chapter 2, we investigate the problem of minimizing the failure rate of packet delivery in the presence of modification attacks and selective forwarding attacks. We propose a novel heuristic approach. This approach is based on randomized multipath routing where the packet initiator generates three copies of the data. Two of them travel towards the base station on two randomized paths separately, and one copy travels through a static

path. This approach consists of two phases: namely, the distributed naming phase and the randomized multipath routing phase. In the distributed naming phase, our approach constructs a shortest path tree and assigns a unique ID to each sensor node. In the second phase, each sensor nodes makes three copies of the sensed data and sends them to the base station via three different paths. We have performed extensive simulations using our network simulator tool, NS2.35, to compare our approach and a state-of-the-art approach proposed in [36]. We use three performance metrics, namely failure rate, network lifetime, latency for comparison. Simulation results show that our approach significantly improves state-of-the-art approach for all the three performance metrics.

In Chapter 3, we investigate the problem of constructing a shortest path overhearing tree with the maximum lifetime. Overhearing can be used to protect the data and investigate the malicious packet modifications or drops. However, the overhearing technique may consume too much energy, shortening the network life time significantly. Consequently, constructing a shortest path overhearing topology with maximum network lifetime is a significant problem. We propose three approaches, a polynomial-time heuristic approach, a MINLP-based approach and an ILP-based approach for both homogeneous WSNs and heterogeneous WSNs.

1. Our polynomial-time heuristic approach uses a priority scheme to find the best parent and monitor to maximize the network lifetime. This main goal of this approach is to minimize the highest energy consumption of all the sensor nodes.

2. Our MINLP-based approach consists of two phases. In the first phase, it partitions all the sensor nodes into m disjoint groups $C_l(l = 1, 2, \cdots, m)$ such that the shortest path length between each sensor node in the group C_l to the base station in the connectivity graph is equal to l. In the second phase, for each sensor node $v_i \in C_l(l = 2, 3, \cdots, m)$, it assigns a monitor and a parent in C_{l-1} to v_i such that the maximum energy consumption per unit time of all the individual sensor nodes is minimized by using MINLP.

3. Our ILP-based approach works in two phases. In the first phase, it partitions all the sensor nodes into m disjoint groups $C_l(l = 1, 2, \cdots, m)$ such that the shortest path length between each sensor node in the group C_l to the base station in the connectivity graph is equal to l. In the second phase, for each group $C_l(l = m, m - 1, \cdots, 2)$, it assigns a monitor and a parent in C_{l-1} to each sensor in C_l such that the maximum energy consumption per time unit of all the individual sensor nodes in C_{l-1} is minimized by using ILP.

We have performed extensive simulation to evaluate the three approaches by using NS 2.35, MIDCO solver, and Intlinprog solver for the heuristic-based, MINLP-based, and ILP-based approaches, respectively. The simulation results show that the polynomial-time heuristic is very effective in increasing the network lifetime compared the ILP-based approach and the MINLP-based approach. One limitation with our shortest path overhearing tree is that if a monitor colludes with the parent of the monitored sensor node, the attacks by the parent of the monitored sensor node may not be detected. We overcome this by selecting a different monitor for every sensor node in every time interval.

In Chapter 4, we investigate reliable and secure end-to-end data aggregation considering selective forwarding attacks and modification attacks in homogeneous cluster-based WSNs. Using concealed end-to-end data aggregation makes it harder for an attacker to eavesdrop on a packet on the air, or modify the data because of the encryption feature.

- In order to aggregate concealed data securely in the presence of both modification attacks and selective forwarding attacks, we propose two data aggregation approaches based on secret sharing, namely, Sign-Share and Sham-Share. However, we found that locating the malicious aggregator in a topology where the packet has to travel through multiple aggregators towards the BS is also significant. Accordingly, we modify our Sign-Share to become Sign-Share Malicious Node Detection.

- In our Sign Share approach, data sensed by a sensor node is split into multiple

shares and sent equally through two aggregators towards the BS. In further detail, the sensed data is encoded, split into shares, each share is encoded, a digital signature is used to sign each share, and then they are sent directly to both aggregators. Aggregators accordingly aggregate the encoded data without understanding the content and transmit directly to the BS.

- Sham-Share is similar to the previous approach, but uses Shamir's secret sharing algorithm. This adds one more feature to the base station: if one aggregator selectively drops or modifies part of the data, the base station will be able to reconstruct the original data.

- In Sign-Share Malicious Node Detection, we replace the digital signature with a Homomorphic Message Authentication Code (MAC), to investigate when and where the data has been violated.

- We have performed extensive simulations using NS3.22 to compare our approaches with the most two relevant approaches, PIP [65] and RCDA-HOMO [67]. The simulation results show that both Sign-Share and Sham-Share consume a reasonable amount of time in processing the data and aggregating the data.

5.2 Future Work

In this section, we discuss several open research problems. The first open problem is to extend our approach to the randomized multipath routing for secure data collection to identify the attacker. Different strategies could be used for an attacker detection. The first strategy is to recover the routing path when the base station does not receive three identical copies. Via analyzing the history of each sensor node, the attacker could be detected. The second strategy is to use overhearing technique. When a sensor node receives a packet and forwards it to another sensor node, a third sensor node will over the

whole process to ensure that no packet is modified or dropped. On-line learning may also be used to detect an attacker.

The second open problem is to construct an overhearing topology for mobile wireless sensor networks such that the network lifetime is maximized. In a mobile network, the location of each sensor node may change constantly. Therefore, it is not feasible to choose a static parent and monitor for each mobile sensor node. New algorithm is needed to dynamically assign a parent and a monitor to each mobile sensor node by considering the sensor mobility and availability.

The third open problem is to construct an overhearing topology that maximizes the network lifetime in a wireless sensor network with multiple base stations. In this case, we need to partition the whole network into disjoint groups such that all the sensor nodes in each group send their data to the designated base station. Efficient distributed algorithms for partitioning the network into disjoint groups and constructing an efficient overhearing topology for each group are needed.

The last open problem is secure data aggregation for clusters with a specific routing topology. In a large WSN, due to the power limitation of transmitter, an aggregator may not send its data to the base station directly. Efficient algorithms are required to ensure sensed data are sent to the base station securely. A tree topology or DAG (Directed Acyclic Graph) could be used for routing. A good algorithm for secure data aggregation needs to not only ensure the security of sensed data, but also be network lifetime aware.

Bibliography

[1] Y. S. E. C. I.F. Akyildiz, W. Su*, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 2, pp. 393–422, 2002.

[2] A. Diaz and P. Sanchez, "Simulation of attacks for security in wireless sensor network," *Sensors*, vol. 16, no. 11, p. 1932, 2016.

[3] J. P. Walters, Z. Liang, W. Shi, and V. Chaudhary, "Wireless sensor network security: A survey," *Security in distributed, grid, mobile, and pervasive computing*, vol. 1, 2007.

[4] A.-S. K. Pathan, H.-W. Lee, and C. S. Hong, "Security in wireless sensor networks: issues and challenges," in *Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference*, vol. 2, pp. 6–pp, IEEE, 2006.

[5] V. Rathod and M. Mehta, "Security in wireless sensor network: a survey," *Ganpat University Journal Of Engineering & Technology*, vol. 1, no. 1, pp. 35–44, 2011.

[6] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[7] A. Perrig, J. Stankovic, and D. Wagner, "Security in wireless sensor networks," *Communications of the ACM*, vol. 47, no. 6, pp. 53–57, 2004.

[8] H. Redwan and K.-H. Kim, "Survey of security requirements, attacks and network integration in wireless mesh networks," in *New Technologies, Mobility and Security, 2008. NTMS'08.*, pp. 1–5, IEEE, 2008.

[9] H. Ng, M. Sim, and C. Tan, "Security issues of wireless sensor networks in healthcare applications," *BT Technology Journal*, vol. 24, no. 2, pp. 138–144, 2006.

[10] Y. Wang, G. Attebury, and B. Ramamurthy, "A survey of security issues in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 8, no. 2, pp. 2–23.

[11] H. Modares, R. Salleh, and A. Moravejosharieh, "Overview of security issues in wireless sensor networks," in *Computational Intelligence, Modelling and Simulation (CIMSiM), 2011 Third International Conference on*, pp. 308–311, IEEE, 2011.

[12] A. Liu, Z. Zheng, C. Zhang, Z. Chen, and X. Shen, "Secure and energy-efficient disjoint multipath routing for wsns," *Vehicular Technology, IEEE Transactions on*, vol. 61, no. 7, pp. 3255–3265, 2012.

[13] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," *Ad hoc networks*, vol. 1, no. 2, pp. 293–315, 2003.

[14] E. Shi and A. Perrig, "Designing secure sensor networks," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 38–43, 2004.

[15] C. Ozturk, Y. Zhang, and W. Trappe, "Source-location privacy in energy-constrained sensor network routing," in *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, pp. 88–93, ACM, 2004.

[16] W. Y. Alghamdi, H. Wu, J. Fei, and S. S. Kanhere, "Randomised multipath routing for secure data collection," in *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on*, pp. 1–6, IEEE, 2014.

[17] R. Maheshwari, J. Gao, and S. R. Das, "Detecting wormhole attacks in wireless networks using connectivity information," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pp. 107–115, IEEE, 2007.

[18] H. W. Jingjing Fei and Y. Wang, "k-dag based lifetime aware data collection in wireless sensor networks," *International Journal of Wireless and Mobile Networks*, vol. 5, no. 5, 2013.

[19] J. Sen, "A survey on wireless sensor network security," *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 1, no. 2, pp. 55–78, 2009.

[20] S. K. Singh, M. Singh, and D. Singhtise, "A survey on network security and attack defense mechanism for wireless sensor networks," *International Journal of Computer Trends and Technology*, vol. 4, no. 2, pp. 1–9, 2011.

[21] K. Vishnu and A. J. Paul, "Detection and removal of cooperative black/gray hole attack in mobile ad hoc networks," *International Journal of Computer Applications*, vol. 1, no. 22, pp. 38–42, 2010.

[22] C. Wang, T. Feng, J. Kim, G. Wang, and W. Zhang, "Catching packet droppers and modifiers in wireless sensor networks," in *Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON'09. 6th Annual IEEE Communications Society Conference on*, vol. 23, pp. 835–843, IEEE, 2012.

[23] K.-F. Ssu, C.-H. Chou, and L.-W. Cheng, "Using overhearing technique to detect malicious packet-modifying attacks in wireless sensor networks," *Computer Communications*, vol. 30, no. 11, pp. 2342–2352, 2007.

[24] B. Bates, A. Keating, and R. Kinicki, "Energy analysis of four wireless sensor network mac protocols," in *Wireless and Pervasive Computing (ISWPC), 2011 6th International Symposium on*, pp. 1–6, IEEE, 2011.

[25] S. Mohammadi and H. Jadidoleslamy, "A comparison of link layer attacks on wireless sensor networks," *International journal on applications of graph theory in wireless ad hoc networks and sensor networks (GRAPH-HOC)*, vol. 3, no. 1, pp. 35–65, 2011.

[26] M. Radi, B. Dezfouli, K. A. Bakar, and M. Lee, "Multipath routing in wireless sensor networks: Survey and research challenges," *Sensors*, vol. 12, no. 1, pp. 650–685, 2012.

[27] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-resilient, energy-efficient multipath routing in wireless sensor networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 4, pp. 11–25, 2001.

[28] H. Hassanein and J. Luo, "Reliable energy aware routing in wireless sensor networks," in *Dependability and Security in Sensor Networks and Systems, 2006. DSSNS 2006. Second IEEE Workshop on*, pp. 54–64, IEEE, 2006.

[29] W. Lou and Y. Kwon, "H-spread: a hybrid multipath scheme for secure and reliable data collection in wireless sensor networks," *Vehicular Technology, IEEE Transactions on*, vol. 55, no. 4, pp. 1320–1330, 2006.

[30] W. Lou, "An efficient n-to-1 multipath routing protocol in wireless sensor networks," in *Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on*, pp. 672–680, IEEE, 2005.

[31] Z. Wang, E. Bulut, and B. K. Szymanski, "Energy efficient collision aware multipath routing for wireless sensor networks," in *Communications, 2009. ICC'09. IEEE International Conference on*, pp. 91–95, IEEE, 2009.

[32] M. Radi, B. Dezfouli, S. A. Razak, and K. A. Bakar, "Liemro: a low-interference energy-efficient multipath routing protocol for improving qos in event-based wireless sensor networks," in *Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on*, pp. 551–557, IEEE, 2010.

[33] S. Li, R. K. Neelisetti, C. Liu, and A. Lim, "Efficient multi-path protocol for wireless sensor networks," *International Journal of Wireless and Mobile Networks*, vol. 2, no. 1, pp. 110–130, 2010.

[34] Y. Ming Lu and V. WS Wong, "An energy-efficient multipath routing protocol for wireless sensor networks," *International Journal of Communication Systems*, vol. 20, no. 7, pp. 747–766, 2007.

[35] J. Deng, R. Han, and S. Mishra, "Countermeasures against traffic analysis attacks in wireless sensor networks," in *Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005. First International Conference on*, pp. 113–126, IEEE, 2005.

[36] T. Shu, M. Krunz, and S. Liu, "Secure data collection in wireless sensor networks using randomized dispersive routes," *Mobile Computing, IEEE Transactions on*, vol. 9, no. 7, pp. 941–954, 2010.

[37] A. M. El-Semary and M. M. Abdel-Azim, "New trends in secure routing protocols for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, 2013.

[38] S. P. Srinivasan and C. Chellappan, "Semi-randomised propagation for secure routing in wireless sensor networks," in *Recent Trends in Information Technology (ICRTIT), 2011 International Conference on*, pp. 428–432, IEEE, 2011.

[39] J. Lee, K. Kapitanova, and S. H. Son, "The price of security in wireless sensor networks," *Computer Networks*, vol. 54, no. 17, pp. 2967–2978, 2010.

[40] L. Dijun, Z. Xiaojun, W. Xiaobing, and C. Guihai, "Maximizing lifetime for the shortest path aggregation tree in wireless sensor networks," in *Proceedings IEEE on INFOCOM 2011*, pp. 1566–1574.

[41] M. Shan, G. Chen, D. Luo, X. Zhu, and X. Wu, "Building maximum lifetime shortest path data aggregation trees in wireless sensor networks," *ACM Transactions on Sensor Networks.*, vol. 11, pp. 11:1–11:24, July 2014.

[42] H. Nabizadeh and M. Abbaspour, "Ifrp: an intrusion/fault tolerant routing protocol for increasing resiliency and reliability in wireless sensor networks," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 14, no. 1, pp. 52–69, 2013.

[43] Y. Hu, Y. Wu, and H. Wang, "Detection of insider selective forwarding attack based on monitor node and trust mechanism in wsn," *Wireless Sensor Network*, vol. 6, no. 11, pp. 237 – 248, 2014.

[44] G.-W. Lee and E.-N. Huh, "Reliable data transfer using overhearing for implicit ack," in *ICCAS-SICE, 2009*, pp. 1976–1979, IEEE, 2009.

[45] Y. Iima, A. Kanzaki, T. Hara, and S. Nishio, "Overhearing-based data transmission reduction for periodical data gathering in wireless sensor networks," in *Proceedings*

of International Conference on Complex, Intelligent and Software Intensive Systems (CISIS'09), pp. 1048–1053, IEEE, 2009.

[46] H.-C. Le, H. Guyennet, and V. Felea, "Obmac: an overhearing based mac protocol for wireless sensor networks," in *Proceedings of International Conference on Sensor Technologies and Applications*, pp. 547–553, IEEE, 2007.

[47] A. Kanzaki, Y. Iima, T. Hara, and S. Nishio, "Overhearing-based data transmission reduction using data interpolation in wireless sensor networks," in *Proceedings of The 5th International conference on Mobile Computing and Ubiquitious Networking*, 2010.

[48] Y. Iima, A. Kanzaki, T. Hara, and S. Nishio, "An evaluation of overhearing-based data transmission reduction in wireless sensor networks," in *Proceedings of Tenth International Conference on Mobile Data Management: Systems, Services and Middleware (MDM'09)*, pp. 519–524, IEEE, 2009.

[49] J. Liang, J. Wang, and J. Chen, "An overhearing-based scheme for improving data persistence in wireless sensor networks," in *Proceedings of 2010 IEEE International Conference on Communications (ICC)*, pp. 1–5, IEEE, 2010.

[50] W. Yan, S. Fahmy, and N. B. Shroff, "On the construction of a maximum-lifetime data gathering tree in sensor networks: Np-completeness and approximation algorithm," in *Proceedings of INFOCOM 2008*, pp. 356–360, 2008.

[51] L. Weifa and L. Yuzhen, "Online data gathering for maximizing network lifetime in sensor networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 1, pp. 2–11, 2007.

[52] C. Yi-Jing, T. Chu-Ping, H. Chih-Hung, L. Kuo-Chi, O. Cheng-Shiou, Y. Chung-Wei, J. Joe-Air, W. Yung-Chung, T. Chwan-Lu, and Y. En-Cheng, "Application of load-balanced tree routing algorithm with dynamic modification to centralized wireless sensor networks," in *IEEE Sensors*, pp. 1392–1395, 2009.

[53] C. Yi-Jing, T. Chu-Ping, L. Kuo-Chi, W. Yung-Cheng, L. Fu-Ming, J. Joe-Air, W. Yung-Chung, T. Chwan-Lu, Y. En-Cheng, and H. Kun-Yaw, "The first order

load-balanced algorithm with static fixing scheme for centralized wsn system in outdoor environmental monitoring," in *IEEE Sensors, 2009*, pp. 1810–1813, 2009.

[54] C. Tzu-Ping, L. Tzu-Shiang, Z. Xiang-Yao, Y. Ping-Lang, and J. Joe-Air, "A load balancing algorithm based on probabilistic multi-tree for wireless sensor networks," in *Proceedings of Fifth International Conference on Sensing Technology (ICST)*, pp. 527–532, 2011.

[55] C. Chia-Pang, W. Jiing-Yi, C. Cheng-Long, L. Tzu-Yun, and J. Joe-Air, "A probablistic load-balancing convergecast tree algorithm for heterogeneous wireless sensor networks," in *Proceedings of IEEE 14th International Conference on High Performance Computing and Communication 2012*, pp. 1624–1628, 2012.

[56] W. Bechkit, M. Koudil, Y. Challal, A. Bouabdallah, B. Souici, and K. Benatchba, "A new weighted shortest path tree for convergecast traffic routing in wsn," in *Proceedings of IEEE Symposium on Computers and Communications (ISCC), 2012*, pp. 187–192, 2012.

[57] L. Junbin, W. Jianxin, C. Jiannong, C. Jianer, and L. Mingming, "An efficient algorithm for constructing maximum lifetime tree for data gathering without aggregation in wireless sensor networks," in *Proceedings of IEEE INFOCOM 2010*, pp. 1–5, 2010.

[58] D. Hui and R. Han, "A node-centric load balancing algorithm for wireless sensor networks," in *IEEE Global Telecommunications Conference (GLOBECOM '03), 2003*, vol. 1, pp. 548–552, 2003.

[59] S. K. A. Imon, A. Khan, M. Di Francesco, and S. K. Das, "Rasmalai: A randomized switching algorithm for maximizing lifetime in tree-based wireless sensor networks," in *Proceedings of INFOCOM 2013*, pp. 2913–2921, IEEE, 2013.

[60] J. Fei, H. Wu, and W. Y. Alghamdi, "Lifetime and latency aware data collection based on k-tree," in *Proceedings of Tenth IEEE International Conference on Intelligent Sensors, Sensor Networks and Information Processing, ISSNIP 2015, Singapore, April 7-9, 2015*, pp. 1–6, 2015.

[61] A. Ouksel and D. Lundquist, "Reducing redundant data transmissions in wireless ad hoc networks: comparing aggregation and filtering," *Wireless Networks*, vol. 21, no. 7, pp. 2155–2168, 2015.

[62] M. Rezvani, A. Ignjatovic, E. Bertino, and S. Jha, "Secure data aggregation technique for wireless sensor networks in the presence of collusion attacks," *Dependable and Secure Computing, IEEE Transactions on*, vol. 12, no. 1, pp. 98–110, 2015.

[63] M. Allahbakhsh and A. Ignjatovic, "An iterative method for calculating robust rating scores," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 26, no. 2, pp. 340–350, 2015.

[64] J. Jose, J. Jose, and M. Princy, "A survey on privacy preserving data aggregation protocols forwireless sensor networks," *CIT. Journal of Computing and Information Technology*, vol. 22, no. 1, pp. 1–20, 2014.

[65] V. Kumar and S. Madria, "Pip: Privacy and integrity preserving data aggregation in wireless sensor networks," in *Reliable Distributed Systems (SRDS), 2013 IEEE 32nd International Symposium on*, pp. 10–19, IEEE, 2013.

[66] Q. Zhou, G. Yang, and L. He, "An efficient secure data aggregation based on homomorphic primitives in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2014, 2014.

[67] C.-M. Chen, Y.-H. Lin, Y.-C. Lin, and H.-M. Sun, "Rcda: recoverable concealed data aggregation for data integrity in wireless sensor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 23, no. 4, pp. 727–734, 2012.

[68] Y.-H. Lin, S.-Y. Chang, and H.-M. Sun, "Cdama: concealed data aggregation scheme for multiple applications in wireless sensor networks," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 25, no. 7, pp. 1471–1483, 2013.

[69] K.-A. Shim and C.-M. Park, "A secure data aggregation scheme based on appropriate cryptographic primitives in heterogeneous wireless sensor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 26, no. 8, pp. 2128–2139, 2015.

[70] B. Przydatek, D. Song, and A. Perrig, "Sia: Secure information aggregation in sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, pp. 255–265, ACM, 2003.

[71] Y. Yang, X. Wang, S. Zhu, and G. Cao, "Sdap: A secure hop-by-hop data aggregation protocol for sensor networks," *ACM Transactions on Information and System Security (TISSEC)*, vol. 11, no. 4, p. 18, 2008.

[72] S. Papadopoulos, A. Kiayias, and D. Papadias, "Exact in-network aggregation with integrity and confidentiality," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 10, pp. 1760–1773, 2012.

[73] P. Haghani, P. Papadimitratos, M. Poturalski, K. Aberer, and J.-P. Hubaux, "Efficient and robust secure aggregation for sensor networks," in *Secure Network Protocols, 2007. NPSec 2007. 3rd IEEE Workshop on*, pp. 1–6, IEEE, 2007.

[74] G. Taban and V. D. Gligor, "Efficient handling of adversary attacks in aggregation applications," in *European Symposium on Research in Computer Security*, pp. 66–81, Springer, 2008.

[75] K. Parmar and D. C. Jinwala, "Malleability resilient concealed data aggregation in wireless sensor networks," *Wireless Personal Communications*, vol. 87, no. 3, pp. 971–993, 2016.

[76] S. Su and H. Yu, "Minimizing tardiness in data aggregation scheduling with due date consideration for single-hop wireless sensor networks," *Wireless Networks*, vol. 21, no. 4, pp. 1259–1273, 2015.

[77] Y. Lu, I.-S. Comsa, P. Kuonen, and B. Hirsbrunner, "Adaptive data aggregation with probabilistic routing in wireless sensor networks," *Wireless Networks*, pp. 1–15, 2015.

[78] Q. Zhou, G. Yang, and L. He, "A secure-enhanced data aggregation based on ecc in wireless sensor networks," *Sensors*, vol. 14, no. 4, pp. 6701–6721, 2014.

[79] E. Mykletun, J. Girao, and D. Westhoff, "Public key based cryptoschemes for data concealment in wireless sensor networks," in *Communications, 2006. ICC'06. IEEE International Conference on*, vol. 5, pp. 2288–2295, IEEE, 2006.

[80] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *Advances in cryptologyEUROCRYPT 2003*, pp. 416–432, Springer, 2003.

[81] S. Ozdemir and H. Çam, "Integration of false data detection with data aggregation and confidential transmission in wireless sensor networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 18, no. 3, pp. 736–749, 2010.

[82] S. Ozdemir and Y. Xiao, "Integrity protecting hierarchical concealed data aggregation for wireless sensor networks," *Computer Networks*, vol. 55, no. 8, pp. 1735–1746, 2011.

[83] D. Westhoff, J. Girao, and M. Acharya, "Concealed data aggregation for reverse multicast traffic in sensor networks: Encryption, key distribution, and routing adaptation," *Mobile Computing, IEEE Transactions on*, vol. 5, no. 10, pp. 1417–1431, 2006.

[84] C. Castelluccia, E. Mykletun, and G. Tsudik, "Efficient aggregation of encrypted data in wireless sensor networks," in *Mobile and Ubiquitous Systems: Networking and Services, 2005. MobiQuitous 2005. The Second Annual International Conference on*, pp. 109–117, IEEE, 2005.

[85] W. He, X. Liu, H. Nguyen, K. Nahrstedt, and T. Abdelzaher, "Pda: Privacy-preserving data aggregation in wireless sensor networks," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pp. 2045–2053, IEEE, 2007.

[86] T. Claveirole, M. D. De Amorim, M. Abdalla, and Y. Viniotis, "Securing wireless sensor networks against aggregator compromises," *Communications Magazine, IEEE*, vol. 46, no. 4, pp. 134–141, 2008.

[87] H. Sethi, D. Prasad, and R. Patel, "Eirda: An energy efficient interest based reliable data aggregation protocol for wireless sensor networks," *International Journal of Computer Applications*, vol. 22, no. 7, pp. 20–25, 2011.

[88] H. Hayouni and M. Hamdi, "Secure data aggregation with homomorphic primitives in wireless sensor networks: A critical survey and open research issues," in *Networking, Sensing, and Control (ICNSC), 2016 IEEE 13th International Conference on*, pp. 1–6, IEEE, 2016.

[89] W. Min, C. Ruixiang, and H. Shunbin, "A secure data aggregation approach in hierarchical wireless sensor networks," in *Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication*, p. 89, ACM, 2016.

[90] X. Li, D. Chen, C. Li, and L. Wang, "Secure data aggregation with fully homomorphic encryption in large-scale wireless sensor networks," *Sensors*, vol. 15, no. 7, pp. 15952–15973, 2015.

[91] M. B. O. Rafik and F. Mohammed, "Sa-spkc: Secure and efficient aggregation scheme for wireless sensor networks using stateful public key cryptography," in *Programming and Systems (ISPS), 2013 11th International Symposium on*, pp. 96–102, IEEE, 2013.

[92] A. Jorio, S. El Fkihi, B. Elbhiri, and D. Aboutajdine, "An energy-efficient clustering routing algorithm based on geographic position and residual energy for wireless sensor network," *Journal of Computer Networks and Communications*, vol. 2015, 2015.

[93] C. Nie, H. Wu, and W. Zheng, "Latency and lifetime-aware clustering and routing in wireless sensor networks," in *Local Computer Networks (LCN), 2016 IEEE 41st Conference on*, pp. 164–167, IEEE, 2016.

[94] C. Nie and H. Wu, "Lifetime-aware clustering and dag-based routing in wsns," in *Proceedings of the 12th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, pp. 77–86, ACM, 2016.

[95] S. Agrawal and D. Boneh, "Homomorphic macs: Mac-based integrity for network coding," in *International Conference on Applied Cryptography and Network Security*, pp. 292–305, Springer, 2009.

[96] S. Mahmud and H. Wu, "Lifetime aware deployment of k base stations in wsns," in *Proceedings of the 15th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*, pp. 89–98, ACM, 2012.

Abbreviations

WSN Wireless Sensor Network

BS Base Station

ILP Integer Linear Programming

MINLP Mixed Integer Non-Linear Programming

RCDA-Homo Recoverable Concealed Data Aggregation- Homogeneous

NS Network Simulator

SMRP Secure Multipath Routing Protocol

MTRP Multicast-Tree Random Propagation

CDA Concealed Data Aggregation

Sign-Share Signature-Secret Sharing

Sham-Share Shamir's Secret Sharing

DAG Directed Acyclic Graph

HMAC Homomorphic Message Authentication Code