Detecting Sybil Attack in Mobile Wireless Sensor Networks Using Observer Nodes

By:

ABDULAZIZ ABDULRAHMAN ALOMIRAH

Supervisor:

Mr. BAHMAN SASSANI (SARAFPOUR)

A thesis submitted in partial fulfilment of the requirements for the degree of Master of Computing

Unitec Institute of Technology 2019
Abstract

Wireless sensor network (WSN) is one of the dominant technology trends in recent years. These networks are being employed in different environments to gather data for various applications. Mobile wireless sensor network (MWSN) is a subclass of WSN, in which the nodes are mobile and frequently change their locations.

Since sensor nodes have limited capabilities, WSNs are vulnerable to various types of attacks, one of which is Sybil attack. In this attack, a malicious node illegitimately forges several (fake) identities. These fake copies confuse and collapse the network. Sybil attack causes too many threats to the routing algorithm, data aggregation, fair resource allocation, voting system, and misbehaviour detection. Since multiple copies of the malicious nodes can be located in several places at the same time, Sybil can disrupt geographic routing protocols and collide with routing algorithms by building many routes from a single node. As a result, detecting and preventing this type of attack is crucial for the security of the wireless sensor network.

In this thesis, I propose a new lightweight algorithm for detecting the Sybil attack in MWSN using observer nodes. Observer nodes are normal, trustful nodes which have been initially programmed to observe the network and report malicious behaviours. An observer node counts the number of times a node has appeared as a common neighbour between itself and its neighbours. After collecting some information about its neighbours, each observer node considers the nodes whose counters are above a threshold as critical, and nodes having all critical nodes in their neighbourhood are considered suspicious nodes.

The results show that true detection rate of the proposed algorithm is 98.1%, and its false detection rate is 0.5%, while similar algorithms could not achieve better than 95.4% and 1.2% for these metrics, respectively. In addition, the proposed algorithm outperforms other algorithms in terms of overhead and scalability.

Keywords:

Acknowledgements

Firstly, I would like to express sincere gratitude to principle, Mr. Bahman Sassani (Sarrafpour) under whose supervision I have completed my thesis at Unitec Institute of technology. My research would have been incomplete without his ultimate supervision, guidance, and support and I am extremely grateful for that. I am glad to have worked with him as my thesis supervisor. His shared visions have helped me throughout the thesis.

I also would like to thank my second supervisor Ms. Xiaosong Li for all her help and feedback given. I owe so much to her for she has always been honest about her opinions. Her valuable comments have obliged a lot to gain improvements in my thesis.

Another credit of my success goes to my sister Reem, she stood with me like a complete support system. My parents have always believed in me and supported me throughout my years of study. The love and care of my family have always kept me inspired to work harder than ever and to be able to complete the process of researching. This accomplishment would not have been possible without any of these people.

Last but not least, I would like to thank the Saudi Arabian Cultural Mission (SACM) and the Ministry of Higher Education in Saudi Arabia for their continuous support and for providing me with this great opportunity to pursue my dreams and achieve my goals.

God bless you all.

Abdulaziz Alomirah
Unitec Institute of Technology
January 2019
# Table of Contents

Acknowledgements .......................................................................................................................... iii

1  Table of Contents ..................................................................................................................... iv

2  List of Figures .......................................................................................................................... vii

3  List of Tables ............................................................................................................................ ix

0  List of Abbreviations ................................................................................................................ x

1  Chapter One: Introduction ....................................................................................................... 1
   1.1  Problem Definition .............................................................................................................. 3
   1.2  Research Motivation .......................................................................................................... 3
   1.3  Research Questions ........................................................................................................... 4
   1.4  Research Contribution ...................................................................................................... 4
   1.5  Thesis Layout .................................................................................................................... 4

2  Chapter Two: Background ....................................................................................................... 5
   2.1  Wireless Sensor Network .................................................................................................. 5
      2.1.1  Components of WSN ............................................................................................... 5
      2.1.2  Characteristics of WSN .......................................................................................... 7
   2.2  WSN Routing Protocols .................................................................................................. 8
   2.3  Attacks in Wireless Sensor Networks ............................................................................. 9
      2.3.1  Classification of Attacks in WSN ............................................................................ 10
   2.4  Summary .......................................................................................................................... 13

3  Chapter Three: Sybil Attack .................................................................................................... 14
   3.1  Types of Sybil Attack ....................................................................................................... 14
   3.2  Detecting Sybil Attack .................................................................................................... 14
      3.2.1  RSSI-Based Methods ............................................................................................. 15
      3.2.2  TDOA Method ....................................................................................................... 17
      3.2.3  Location-Based Methods ....................................................................................... 17
      3.2.4  AOA-Based Methods ............................................................................................. 18
3.2.5 Distributed Methods ............................................................................. 19
3.2.6 Using Machine Learning Algorithms for Detecting Sybil Attack .......... 19
3.2.7 Authentication-Based Methods .............................................................. 20
3.2.8 Detecting Sybil Attack in Mobile Wireless Sensor Networks ............. 21
3.3 Summary .................................................................................................. 26

4 Chapter Four: Methodology ...................................................................... 29
4.1 Hypothesis ............................................................................................... 29
4.2 Research Approach .................................................................................. 29
4.3 System Assumptions and Attack Model .................................................. 30
4.4 Proposed Algorithm ................................................................................ 31
4.5 Simulation Setup ...................................................................................... 33
4.6 Data Collection Process ......................................................................... 36
4.7 Summary .................................................................................................. 36

5 Chapter Five: Evaluation of the Proposed Algorithm ............................. 37
5.1 Data Collection Process ......................................................................... 37
5.2 Evaluation metrics ................................................................................... 37
5.3 The Complexity of the Proposed Algorithm .......................................... 38
  5.3.1 Memory Overhead ............................................................................. 38
  5.3.2 Communication Overhead ................................................................. 38
  5.3.3 Computation Overhead ..................................................................... 39
5.4 Detection Rates for Different Rounds of the Algorithm ......................... 40
5.5 Impact of Various Factors on Detection Rates ....................................... 40
  5.5.1 Number of Sybil Identifiers Forged by Malicious Nodes ................. 40
  5.5.2 Number of Malicious Nodes ............................................................... 42
  5.5.3 Number of Observer Nodes ............................................................... 44
  5.5.4 Threshold .......................................................................................... 49
  5.5.5 Transmission Range ......................................................................... 50
List of Figures

Figure 1-1 Applications of WSN (Agrawal, 2018)  1
Figure 1-2 Sybil Attack (Zamani & Jafri, 2014)  2
Figure 2-1. A sample sensor node (Libelium, 2013)  6
Figure 2-2. The structure of a sensor node (AboZahhad, Farrag, & Ali, 2015)  6
Figure 2-3. Gateway node for WSN (Dener, 2016)  7
Figure 2-4. AODV routing mechanism (Iqbal, et al., 2014)  8
Figure 2-5. Passive attack in WSN Shabana et al., (2016)  10
Figure 2-6. Black hole attack (Shabana, Fida, Khan, Jan, & Rehman, 2016)  11
Figure 2-7. Jamming in WSN (Shabana, Fida, Khan, Jan, & Rehman, 2016)  11
Figure 2-8. Replay attack in WSN  12
Figure 3-1 Categorisation of Sybil attack detection algorithms in WSNs (Shehni, Faez, Farshad, & Kelarestaghi, 2017)  15
Figure 3-2. A sample scenario in (Demirbas & Song, 2006)  16
Figure 3-3. Triangulation method (Tangpong, 2010)  18
Figure 3-4. Social network model of Sybil attack (Zeng & Chen, 2010)  19
Figure 3-5. Minimising the edges between the honest region and the Sybil region (Zeng & Chen, 2010)  20
Figure 3-6. An example of node locations in WSN Jamshidi et al., (2017).  22
Figure 3-7. The moving history of W1 in Figure 3-7 (Jamshidi, Zangeneh, Esnaashari, & Meybodi, 2017).  23
Figure 3-8. Location of nodes after the second round of the algorithm (Jamshidi, Zangeneh, Esnaashari, & Meybodi, 2017).  23
Figure 3-9. Left: Moving_history of W1; Right: Moving_history of W3 after the second round (Jamshidi, Zangeneh, Esnaashari, & Meybodi, 2017).  24
Figure 3-10 Data structures used in (Shehni, Faez, Farshad, & Kelarestaghi, 2017):  24
Figure 3-11. The state diagram for calculating elements of $C^k_{co-pres}$ (Shehni, Faez, Farshad, & Kelarestaghi, 2017)  25
Figure 3-12. The true detection rate of several Sybil detection algorithms  27
Figure 3-13. The false detection rate of several Sybil detection algorithms  27
Figure 4-1. A malicious node which forges four IDs  31
Figure 4-2. The idea of the proposed algorithm  32
Figure 4-3. Two Ray Ground Propagation (Eenennaam, 2008)  34
Figure 4-4. Simulator Architecture
Figure 5-1. Detection rate of the proposed algorithm in different rounds.
Figure 5-2. The true detection rate for various number of Sybil IDs
Figure 5-3. The false detection rate for various numbers of Sybil IDs
Figure 5-4. The true detection rate for various numbers of malicious nodes
Figure 5-5. The false detection rate for various numbers of malicious nodes
Figure 5-6. The true detection rate for various number of observer nodes
Figure 5-7. The false detection rate for various number of observer nodes
Figure 5-8. The true detection rate for various numbers of observer nodes and malicious nodes
Figure 5-9. The false detection rate for various numbers of observer nodes and malicious nodes
Figure 5-10. The true detection rate for various numbers of observer nodes and Sybil IDs
Figure 5-11. The false detection rate for various numbers of observer nodes and Sybil IDs
Figure 5-12. The true detection rate for different values of theta
Figure 5-13. The false detection rate for different values of theta
Figure 5-14. True/false detection rate for different transmission ranges
Figure 5-15. The true/false detection rate for various numbers of nodes
Figure 5-16. The true detection rate for various numbers of nodes and number of observer nodes
Figure 5-17. The false detection rate for various numbers of nodes and number of observer nodes
Figure 5-18. The true detection rate for various numbers of nodes and the threshold
Figure 5-19. The false detection rate for various numbers of nodes and the threshold
Figure 5-20. The true detection rate for various numbers of nodes and Sybil IDs
Figure 5-21. The false detection rate for various numbers of nodes and Sybil IDs
Figure 5-22. The true detection rate for various numbers of nodes and malicious nodes
Figure 5-23. The false detection rate for various numbers of nodes and malicious nodes
Figure 5-24. Comparing the true detection rate of the proposed algorithm with similar researches
Figure 5-25. Comparing the false detection rate of the proposed algorithm with similar researches
Figure 5-26. Comparing the scalability of the proposed algorithm with similar researches
List of Tables

Table 3-1. Comparing the complexity of several Sybil detection algorithms (Yu, Lu, & Kuo, 2008) 26
Table 5-1. Network configuration for simulation. 37
Table 5-2. Comparing the complexity of the proposed algorithm with (Jamshidi, Zangeneh, Esnaashari, & Meybodi, 2017) 39
Table 5-3. Network configuration for investigating the impact of the number of Sybil IDs 41
Table 5-4. Network configuration for investigating the impact of the number of malicious nodes 42
Table 5-5. Network configuration for investigating the impact of the number of observer nodes 44
Table 5-6. Network configuration for investigating the impact of the number of observer nodes and malicious nodes 45
Table 5-7. Network configuration for investigating the impact of the number of observer nodes and Sybil IDs 47
Table 5-8. Network configuration for investigating the impact of theta 49
Table 5-9. Network configuration for investigating the impact of transmission range 50
Table 5-10. Network configuration for investigating the impact of the number of nodes 51
Table 5-11. Network configuration for investigating the impact of the number of nodes and number of observer nodes 52
Table 5-12. Network configuration for investigating the impact of the number of nodes and the threshold 54
Table 5-13. Network configuration for investigating the impact of the number of nodes and Sybil IDs 56
Table 5-14. Network configuration for investigating the impact of the number of nodes and malicious nodes 58
Table 5-15. Network configuration for comparing the detection rated of the proposed algorithm with (Jamshidi et al., 2017) and (Shehni et al., 2017) 59
Table 5-16. Network configuration for comparing the scalability of the proposed algorithm with (Jamshidi et al., 2017) and (Shehni et al., 2017) 61
## List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACO</td>
<td>Ant Colony Optimization</td>
</tr>
<tr>
<td>AOA</td>
<td>Angle of Arrival</td>
</tr>
<tr>
<td>AODV</td>
<td>Ad hoc On Demand Distance</td>
</tr>
<tr>
<td>CH</td>
<td>Cluster-Head</td>
</tr>
<tr>
<td>DSDV</td>
<td>Destination-Sequential Distance Vector</td>
</tr>
<tr>
<td>DSR</td>
<td>Dynamic Source Routing</td>
</tr>
<tr>
<td>GEAR</td>
<td>Geographic and Energy-Aware Routing</td>
</tr>
<tr>
<td>LEACH</td>
<td>Low Energy Adaptive Clustering Hierarchy</td>
</tr>
<tr>
<td>MAC</td>
<td>Media Access Control</td>
</tr>
<tr>
<td>MEMS</td>
<td>Micro-electro-mechanical systems</td>
</tr>
<tr>
<td>MWSN</td>
<td>Mobile Wireless Sensor Network</td>
</tr>
<tr>
<td>OLSR</td>
<td>Optimized Link State Routing</td>
</tr>
<tr>
<td>ON</td>
<td>Observer Node</td>
</tr>
<tr>
<td>RPG</td>
<td>Random Password Generation</td>
</tr>
<tr>
<td>RREP</td>
<td>Route Reply</td>
</tr>
<tr>
<td>RREQ</td>
<td>Route Request</td>
</tr>
<tr>
<td>RSP</td>
<td>Received Signal Powers</td>
</tr>
<tr>
<td>RSSI</td>
<td>Received Signal Strength Indication</td>
</tr>
<tr>
<td>SN</td>
<td>Sensor Node</td>
</tr>
<tr>
<td>TDOA</td>
<td>Time Difference of Arrival</td>
</tr>
<tr>
<td>TEBA</td>
<td>Trust Evaluation Base on AOA</td>
</tr>
<tr>
<td>TOA</td>
<td>Time on Arrival</td>
</tr>
<tr>
<td>TRG</td>
<td>Two Ray Ground</td>
</tr>
<tr>
<td>VANET</td>
<td>Vehicular Ad hoc Network</td>
</tr>
<tr>
<td>WRP</td>
<td>Wireless Routing Protocol</td>
</tr>
<tr>
<td>WN</td>
<td>Watchdog Node</td>
</tr>
<tr>
<td>WSN</td>
<td>Wireless Sensor Network</td>
</tr>
<tr>
<td>ZRP</td>
<td>Zone Routing Protocol</td>
</tr>
</tbody>
</table>
Chapter One: Introduction

Micro-electro-mechanical systems (MEMS) and wireless communication technologies witnessed a major breakthrough in the last few years. This has facilitated fabricating small devices that can run autonomously, large-scale, low powered, and inexpensive, which suits the needs of many industries. Such devices can be used in a kind of distributed networking called Wireless Sensor Networks (WSNs). Wireless sensor network (WSN) is a subclass of the ad-hoc wireless network, which consists of hundreds to thousands of sensor nodes spread in an environment to collect data. These networks can be used in many applications, some of which have been shown in Figure 1-1.

![Figure 1-1 Applications of WSN (Agrawal, 2018)](image)

Sensor nodes have limited storage, communication, and processing capabilities. A non-chargeable battery often supplies their energy. Nodes consume energy for sending, receiving, and processing data. When the battery is totally consumed, the node is dead. When the number of dead nodes exceeds a certain number, the coverage of the network becomes weak, and the network may discontinue its normal functions.
Research on various aspects of WSN started from the early 1990s (Athanasios, 2011). One of the major concerns about these networks is their security, especially when they are used in critical applications, such as military missions (Akyildiz & Kasimoglu, 2004). There are many types of known attacks against WSN, among which Sybil attack is one of the most famous and serious (Levine, Shields, & Margolin, 2006). In this attack, an attacker captures a legal node (victim) or inserts an illegal node in the network. The captured node is called the malicious node, which may duplicate the identifier of existing nodes or forge multiple fake identifiers in different areas of the network. A model of the Sybil attack is shown in Figure 1-2. The malicious node forges the following fake IDs: A, B, C, D, E, F, M, N.

![Figure 1-2 Sybil Attack (Zamani & Jafri, 2014)](image)

In this way, many nodes communicate with the malicious node, and, hence, it can considerably affect the traffic and the routing protocols. It can also intervene in other functionalities of the network, such as voting, reputation evaluation, and data aggregation (Newsome., Shi, Song, & Perrig, 2004).

Although the Sybil attack threatens other types of networks, such as Vehicular Ad-hoc Network (VANET) (Kaur, Devgan, & Singh, 2016), according to Palak (2017), WSN is the most vulnerable network for this attack. On the other hand, Sybil is the most dangerous attack on WSN, as it affects the whole network (Palak, 2017).
1.1 Problem Definition

As discussed in the previous section, Sybil is among the most serious attacks against WSN. So far, there have been many approaches proposed to detect the Sybil attack in non-mobile (static) WSN, which include radio resource testing, random key pre-distribution Newsome et al., (2004), received signal strength indication (RSSI) (Demirbas & Song, 2006), neighbourhood information (Yang, Chen, & Trappe, 2008), time difference of arrival (TDOA), and identifying angle of arrival (AOA) (Wen, Li, & Zheng, 2008). However, none of these algorithms can be applied to mobile wireless sensor networks where nodes (including Sybil nodes) are frequently moving and changing their locations.

Mobile wireless sensor networks (MWSN) consist of mobile sensor nodes which can be attached to humans, animals, cars, or any other mobile devices. Detecting a Sybil attack in MWSN is more challenging than in static WSN.

Research conducted on detecting the Sybil attack in MWSN using various techniques included clustering (Sharmila & Umamaheswari, Detection of sybil attack in mobile wireless sensor networks, 2012), validating nodes by the base station (Sharmila & Umamaheswari, Node ID based detection of Sybil attack in mobile wireless sensor network, 2012) and using watchdog nodes Jamshidi et al., (2017). These algorithms either suffer from low accuracy or consist of complicated phases and are unsuitable for MWSN with limited resources.

In this thesis, I propose a new, lightweight, scalable algorithm to detect the Sybil nodes in mobile wireless sensor networks, which addresses the issues with the previous researches. The proposed algorithm does not rely on RSSI, AOA, radio resource testing, or random key pre-distribution. It uses observer nodes, which are normal, trustful nodes in the WSN which have been initially programmed to observe the network and report malicious behaviours. An observer node finds the common neighbours between itself and its neighbours. After calculating the frequency of nodes in the common lists, each observer node considers the nodes appearing more than a threshold as critical nodes. Then, the nodes having all critical nodes in their neighbourhood will be considered suspicious nodes.

1.2 Research Motivation

Security is an important concern regarding wireless sensor networks. One serious threat against such networks is Sybil attack. Existing approaches are not convenient for mobile nodes with limited capabilities. Therefore, the main motivation of this research is proposing a
practical and lightweight algorithm for detecting the Sybil nodes in MWSN which addresses issues with previous algorithms.

1.3 Research Questions
The questions that came to my mind in the initial stage of the research was as follows:

What lightweight, scalable algorithm can be developed to detect the Sybil nodes in MWSNs?

Then, I broke it to more detailed questions as follows:

- How many observer nodes would be enough for a wireless sensor network to detect the Sybil attack?
- How often should an observer node collect information about neighbouring nodes?
- What is a suitable threshold for common neighbours to mark a node as suspicious?
- How much overhead would the observation algorithm impose on the network?

1.4 Research Contribution
In this research, I propose a new algorithm for detecting the Sybil attack in mobile wireless sensor networks using observer nodes for monitoring the network traffic passively, as well as nodes’ mobility to find Sybil nodes accurately. The algorithm is intended to be lightweight by minimising memory, computation, and communication overheads.

1.5 Thesis Layout
The rest of the thesis is organised as follows. Chapter 2 introduces wireless sensor networks, their characteristics, and the known attacks against them. Sybil attack and existing detection algorithms are discussed in Chapter 3. The proposed algorithm is detailed in Chapter 4 and will be evaluated in Chapter 5. Finally, Chapter 6 concludes the thesis and offers several directions for the future research.
Chapter Two: Background

Wireless sensor network (WSN) is a type of ad-hoc wireless networks which consists of a group of sensor nodes spread into an environment to gather data for monitoring and recording the physical conditions. Sensor nodes have limited resources and communicate with each other through a wireless network. In these networks, all the data gathered by the sensor nodes are forwarded to a sink node. Because of resource constraints (i.e., memory, processor, communication), sensor nodes are susceptible to many types of attacks. This is while security is critical for many applications of WSN such as military missions and medical systems. Therefore, it is important to detect and prevent attacks to make WSN secure.

In this chapter, the basic concepts that help the reader to understand the rest of the thesis are explained. More specifically, the chapter briefly introduces WSN, its characteristics and its applications. Then, existing attacks in WSN are reviewed.

2.1 Wireless Sensor Network

A wireless sensor network (WSN) is a type of ad-hoc wireless networks, which can be used in many applications such as military, healthcare, agriculture, environmental monitoring or surveillance, and weather forecasting. Such networks may consist of hundreds to thousands of sensor nodes randomly scattered in an environment. Sensor nodes have limited memory, processor, and transmission range. The energy of a node comes from a non-chargeable battery which, when finished, renders the node dead.

In this section, I review the components, the architecture, and characteristics of WSN.

2.1.1 Components of WSN

WSN consists of the following components:

Sensor node: A sensor node which is also known as a mote, is a device with some sensing capability. Sensor nodes have also limited capabilities to process data and communicate with each other wirelessly. A sample sensor node is shown in Figure 2-1.
The structure of a sensor node is shown in Figure 2-2. As it is shown in the figure, a sensor node consists of three main components: sensing, computing, and communication. Since the cost of sensor nodes is a major concern when deploying a wireless sensor network, their capabilities (processing, memory, communication, and energy) are deliberately constrained to reduce their manufacturing cost.

**Gateway:** A gateway provides a connection between sensor nodes and the application platform. Data collected by wireless nodes is aggregated and manipulated by the gateway and is forwarded to the application platform which may be running on a local computer, a networked computer, or a mobile device.
Actuator: An actuator is responsible for moving and controlling the physical parts of a system. For example, an actuator of a monitoring system for the temperature in a room is responsible for controlling the air-conditioning equipment.

2.1.2 Characteristics of WSN

The main characteristics of WSN which differentiate it from other types of networks are as follows:

- **Power efficiency:** Nodes are usually supplied with non-rechargeable batteries. Thus, efficiency in power consumption is vital for such networks.
- **Fault tolerance:** Nodes are prone to failure. Thus, they must have the ability to continue network functionalities in case of node failures.
- **The mobility of nodes:** Some of the sensor networks consist of mobile nodes. This type of sensor networks is known as mobile sensor networks (MWSN). Nodes are mobile either to increase communication efficiency or due to the type of applications.
- **Heterogeneity of nodes:** In a WSN, there are different types of nodes which must work cooperatively to provide the intended service of the network.
- **Scalability:** In WSN nodes are densely deployed. So, WSN should be scalable to manage a large number of nodes.
- **Responsiveness:** WSN must adapt quickly when the topology of the network changes.
- **Communication failures:** When any node fails to exchange data with other nodes, the base station or gateway node must be quickly informed.
2.2 WSN Routing Protocols

Routing protocols are used by sensor nodes to send data to other nodes or to the base station. There are many routing protocols for WSN of which we briefly introduce the most popular ones in this section. These routing protocols can be classified based on several parameters as follows (DEKIVADIYA, 2012):

- Mode of functioning and applications: The routing protocols can be classified as:
  - **Proactive protocols**: In proactive protocols, which are also referred to as table-driven routing protocols, each node maintains a routing table which consists of routing information to all other nodes. Destination-sequenced Distance Vector (DSDV) (Perkins & Bhagwat, 1994), Optimised Link State Routing (OLSR) (Clausen & Jacquet, 2003), and Wireless Routing Protocol (WRP) (Murthy & Garcia-Luna-Aceves, 1996) are examples of proactive protocols.
  - **Reactive protocols**: In reactive protocols, which are also called on-demand protocols, nodes do not keep routing tables. Instead, they search for the routes when they want to send data to a destination. Ad hoc On-Demand Distance Vector (AODV) (Perkins, Belding-Royer, & Das, 2003) and Dynamic Source Routing (DSR) (Johnson, Hu, & Maltz, 2007) are examples of reactive protocols.

Since AODV protocol has been used for this research, it is explained in more details here. In AODV when a node wants to send data to a destination, it sends the Route Request (RREQ) to its neighbours. The neighbours will forward RREQ to their own neighbours until at least one node finds a route to the destination. Then, it sends back the Route Reply (RREP) with the information about the route to the destination. This process is shown in Figure 2-4.

![Figure 2-4. AODV routing mechanism (Iqbal, et al., 2014)](image-url)
- **Hybrid protocols**: Hybrid protocols are a combination of reactive and proactive protocols. Zone Routing Protocol (ZRP) (Haas & Pearlman, 1998) is an example of hybrid protocols.

- Participation style of nodes: Classifications are as follows:
  - **Direct communication protocols**: In these protocols nodes send data to their destinations through the base station. SPIN (Pattani & Chauhan, 2015) is an example of this type of protocols.
  - **Flat protocols**: In flat protocols, nodes send packets to the base station through multiple intermediate nodes. In other words, the sender finds a route to the base station for sending its data. Rumor (Braginsky & Estrin, 2002) is an example of a flat protocol.
  - **Clustering protocols**: These protocols divide nodes into groups (clusters). Each cluster has a cluster-head (CH), which can directly communicate with the base station. All cluster members must send their data to the CH. Low Energy Adaptive Clustering Hierarchy (LEACH) (Heinzelman, Chandrakasan, & Balakrishnan, 2000) is an example of clustering algorithms.

- Network structure: There are three classes of protocols as follows:
  - **Data centric protocols**: In this type of protocols, attribute-based naming is used to specify the property of data. These properties are used to answer queries for which nodes collect data from specific areas. Collected data is sent to the base station. SPIN is one of the data-centric protocols.
  - **Hierarchical protocols**: These protocols are energy efficient. Nodes with higher energy are responsible for routing and nodes with lower energy are responsible for sending data. Teen (Manjeshwar & Agrawal, 2001) is a hierarchical protocol.
  - **Location based protocols**: These protocols use the location of sensor nodes to find an optimal path from source to destination. The nodes location can be determined using GPS (Global Positioning System) signals, received radio signal strength, etc. Geographic and Energy-Aware Routing (GEAR) (Yu, Estrin, & Govindan, 2001) is a location-based protocol.

2.3 Attacks in Wireless Sensor Networks

As it was discussed in the previous section, sensor nodes have limited resources and are not usually controlled by a central unit. Therefore, providing full security to WSN is a big challenge.
The main goal of the security in WSN is to provide confidentiality, authenticity, integrity, and availability of all messages even when there exist resourceful adversaries. Since WSN is usually deployed in remote places and nodes are left unattended, these networks must be equipped with security mechanisms to defend any attack against the network.

2.3.1 Classification of Attacks in WSN

In Messai (2014), the attacks against WSN are classified as follows:

- **Passive attack (eavesdropping):** These attacks usually monitor the packets exchange within the network and do not have any direct impact on the network. Since the attackers do not interfere with data communication, detection of this type of attacks is difficult. Figure 2-5 depicts this type of attack.

- **Active attack:** Attackers exchange data with some nodes and may disrupt the normal functionality of the network. The most well-known active attacks are as follows:
  - **Tampering:** The attacker has physical access to the node to recover cryptographic material such as the keys used for ciphering.
  - **Black hole attack:** The attacker falsifies the routing information to redirect the traffic towards itself. Then, the packets coming through the traffic to the attacker node are dropped. Figure 2-6 depicts this type of attack.
Selective forwarding: The attacker plays the role of a router and refuses to forward certain messages and simply drops them.

Sybil attack: The attacker forges multiple identities in an illegitimate way. In this way, the attacker can intervene in some algorithms as election, routing, and data aggregation.

HELLO flood attack: Many routing protocols require the nodes to send "HELLO" packet to discover the route and neighbouring nodes. In HELLO flood attack, the attacker floods the network with HELLO messages to prevent other messages from being exchanged.

Jamming: This attack is a kind of Denial of Service (DOS) attack in which the attacker tries to jam the frequency of radio channels used for communication of nodes. This happens by sending useless information on the frequency bands. This jamming can be temporary, intermittent or permanent. Figure 2-7 depicts jamming in WSN.
- **Blackmail attack:** A malicious node announces a legitimate node as malicious which causes the legitimate node eliminated from the network. If the attacker can successfully eliminate many legitimate nodes from the network, the operation of the network may be disrupted.

- **Exhaustion:** The attacker wastes the energy of its victim by forcing it to do unnecessary calculations or to sending/receiving unnecessarily data.

- **Replay attack:** In this attack, an adversary may repeat or delay a message which was already sent to a victim. For example, assume Alice contacts Bob and Bob asks for her identity. Alice sends her password (which may be encrypted) to Bob. If the attacker sniffs the message, it may contact Bob and resend Alice’s encrypted password to Bob. In this case, Bob will think it is Alice who contacted him. This is shown in Figure 2-8.

![Figure 2-8. Replay attack in WSN](image)

- **Wormhole attack:** The attackers are spread in different parts of the network and tunnel messages among themselves to fool nodes to think they are neighbours. In this way, attackers mislead the routing algorithm, which may result in excessive packet dropping.

- **Identity replication attack:** The attacker clones nodes and spreads them in different parts of the network to mislead the routing algorithm. Unlike the Sybil attack, in identity replication attack, the same identity is given to different physical nodes.
2.4 Summary

In this chapter, I introduced wireless sensor networks (WSN), which are a special type of wireless ad-hoc networks and consist of a group of sensor nodes which collaborate with each other to achieve a specific goal. The applications of WSN include (but not limited to): military, healthcare, agriculture, environmental monitoring or surveillance, and weather forecasting.

Nodes in some WSNs are mobile which means they change their location from time to time. This special type of WSN is called mobile wireless sensor network (MWSN).

Sensor nodes have limited storage, communication and processing capabilities. Therefore, they are vulnerable to malicious behaviour of attackers. This is while they are usually employed in safety-critical systems for which security is vital. In response to this need, there have been lots of researches about preserving security in WSN. I reviewed most of the well-known attacks against WSN one of which is Sybil attack. In the Sybil attack, a malicious node forges several fake identities to convince its neighbouring nodes that there are many nodes in their neighbourhood. This can have harmful consequences for many algorithms such as routing, voting and data aggregation.

In the next chapter, I will review some of the researches conducted for detecting and preventing Sybil attack in WSN.
Chapter Three: Sybil Attack

In the previous chapter, I reviewed some of the well-known attacks against WSN one of which is Sybil attack. Sybil attack, which is one of the most popular and harmful attacks against WSN, was first introduced in (Douceur, 2002) for peer-to-peer networks. In this attack, a malicious node illegitimately forges several (fake) identities. These fake copies, which are called Sybil nodes, confuse and collapse the network. This attack can affect routing algorithms, data aggregation, fair resource allocation, voting system, and misbehaviour detection. Since multiple copies of the malicious node can be located in several places at the same time, Sybil attack can disrupt geographic routing protocols and collide with routing algorithms by building many routes from a single node (Gharu, Pawar, & Agarwal, 2017). As a result, detecting and preventing this type of attack is crucial for the security of the wireless sensor network.

In this thesis, I investigate Sybil attack in more details and propose a new algorithm to detect this attack. Therefore, in this chapter, I pay more attention to this attack and review some of the researches on detecting and preventing Sybil attack in WSN.

3.1 Types of Sybil Attack

(Newsome., Shi, Song, & Perrig, 2004) Systematically analysed this attack for wireless sensor networks and classified it as follows:

- **Direct vs. indirect communication**: The Sybil attack may directly communicate with legitimate nodes or, alternatively, it may communicate with legitimate nodes through malicious nodes.

- **Fabricated vs. stolen identities**: Sometimes an attacker may fabricate new identities. For example, if the node ID is represented using a 32-bit integer, the attacker can simply generate a random 32-bit value. However, if there is a mechanism to identify legitimate nodes, the attacker cannot fabricate new IDs. Instead, it steals legitimate nodes IDs.

- **Simultaneity**: The attacker may try all its Sybil identities in the network at once or may use a smaller number of them at any given time.

3.2 Detecting Sybil Attack

Many researchers have been conducted to detect the Sybil attack in WSN. In Shehni et al., (2017) a categorisation for Sybil detection algorithms has been outlined (see Figure 3-1). In this section, some researches which are more related to this research are explained.
3.2.1 RSSI-Based Methods

Received Signal Strength Indication (RSSI) is an indication of the power level received by receiver after the antenna loss. RSSI is measured in an arbitrary unit and the higher the RSSI, the stronger the signal. RSSI can be used as an indication of the distance between the sender and the receiver.

(Demirbas & Song, 2006) introduced an RSSI-based locating scheme to estimate the location of nodes in the network using the proportion of RSSIs from multiple receivers. The nodes with the same location are considered as Sybil nodes. Their scheme uses four location-aware nodes (tracking nodes) to monitor packets sent through the network. In collaboration with each other, tracking nodes locate any node which sends packets. A sample scenario is shown in Figure 3-2 where nodes D1, D2, D3, and D4 are tracking nodes and the Sybil node forges two IDs: S1 and S2. Assuming the Sybil node broadcasts a message at time t1 with its first ID, S1. Tracking nodes record the RSSI of the message they received from S1. All tracking nodes send the RSSI of the message to D1. Let $R_j^k$ denotes the RSSI value of the message sent by node k when received by node j. Then, D1 calculates and stores the ratio of RSSI it receives from other nodes at time t1 as follows:

$$\frac{R_{D1}^{S1}}{R_{D2}^{S1}}, \frac{R_{D1}^{S1}}{R_{D3}^{S1}}, \frac{R_{D1}^{S1}}{R_{D4}^{S1}}$$  (3-1)
If the Sybil node broadcasts another message at time t2 with its second ID (S2), all tracking nodes will send their calculated RSSI to D1 again. Then, D1 calculates the ratios as follows:

\[
\frac{R_{D1}^{S2}}{R_{D2}^{S2}}, \frac{R_{D1}^{S2}}{R_{D3}^{S2}}, \frac{R_{D1}^{S2}}{R_{D4}^{S2}}
\]

Then, D1 checks the ratios with the ones at time t1. If they are equal, it identifies S1 and S2 as the same node (i.e., a Sybil node). The equalities that are checked by D1 are as follows:

\[
\frac{R_{D1}^{S1}}{R_{D2}^{S1}} = \frac{R_{D1}^{S2}}{R_{D2}^{S2}}, \frac{R_{D1}^{S1}}{R_{D3}^{S1}} = \frac{R_{D1}^{S2}}{R_{D3}^{S2}}, \frac{R_{D1}^{S1}}{R_{D4}^{S1}} = \frac{R_{D1}^{S2}}{R_{D4}^{S2}}
\]

Figure 3-2. A sample scenario in (Demirbas & Song, 2006)

In the paper, it is discussed that four tracking nodes are enough to detect the Sybil nodes because all Sybil nodes are located adjacent to each other.

(Chen & Yang, 2010) And (Jangra & Priyanka, 2011) have also proposed an RSSI-based approach to detect the Sybil attacks. The proposed approach used LEACH protocol to cluster the network.

An advanced RSSI-based technique has been proposed in (Mistra & Myneni, 2010) to detect the Sybil nodes while they are regulating their transfer powers.

A Sybil detection algorithm has been proposed in (Shi, Liu, & Zhang, 2015) called LEACHRSSI-ID (LRD) which analyses RSSI-ID tables and uses the information about the remaining energy and density of nodes in a cluster to find the Sybil node.
3.2.2 TDOA Method

The TDOA (Time Difference on Arrival) method is an improvement of the TOA (Time on Arrival) method. In TOA, it is essential to know the timestamps when messages are sent and received by the anchor node (beacon). Therefore, it is necessary to synchronise the time in the whole network. However, the TDOA method, instead of propagation time, takes into account the time difference of signal propagation between anchor nodes. As a result, TDOA techniques do not require strict time synchronisation for the WSN (Mi, hui, Yanfei, & Kefei, 2008).

(Saxena & Sejwar, 2014) Proposed an algorithm based on TDOA localisation method for Sybil attack detection in cluster-based networks. The algorithm can detect the malicious behaviour of both cluster-head nodes and member nodes in a cluster. The algorithm considers the network with the routing protocol LEACH in which nodes are clustered and cluster-heads communicate with cluster members. In each cluster, there is a member node which collaborates with three other member nodes to monitor and detect the attack when the cluster-head is a Sybil node. Assume the cluster-head broadcasts a message with its first ID, H1, and the four monitoring nodes M1, M2, M3, and M4 in the cluster receive the message in times t1, t2, t3, and t4, respectively. M2, M3, and M4 send the arriving time of the message to M1 and M1 calculates the TDOA between the nodes and itself as follows:

\[
d_{2,1}^{H_1} = t_2 - t_1, \quad d_{3,1}^{H_1} = t_3 - t_1, \quad d_{4,1}^{H_1} = t_4 - t_1
\]  

(3-4)

If the cluster-head sends another message using its second ID, H2, and the nodes receive the message at times t'1, t'2, t'3, and t'4, respectively, the same process is carried out to calculate the new TDOA:

\[
d''_{2,1}^{H_1} = t'_2 - t'_1, \quad d''_{3,1}^{H_1} = t'_3 - t'_1, \quad d''_{4,1}^{H_1} = t'_4 - t'_1
\]  

(3-5)

Then, M1 calculates the ratio of TDOA as follows:

\[
\frac{d''_{2,1}^{H_1}}{d''_{3,1}^{H_1}}, \quad \frac{d''_{3,1}^{H_1}}{d''_{4,1}^{H_1}}, \quad \frac{d''_{2,1}^{H_1}}{d''_{4,1}^{H_1}}
\]  

(3-6)

If the ratios are equal, M1 identifies the cluster-head as a Sybil node.

3.2.3 Location-Based Methods

These methods are based on this fact that all Sybil identities belong to the same malicious nodes must be in the same location. Locations are verified using specific methods such as triangulation (Tangpong, 2010).
Figure 3-3 shows the essence of triangulation to locate the malicious node. The scheme needs three trustful nodes to determine the location of other nodes based on the strength of the received signal. To find the location of each node, the three trustful nodes calculate their distances with the node. Then, by solving the following equations they find the intersection of the three circles, which is the location of the node. If several nodes have the same location, it means they are different identifiers forged by a Sybil node.

\[
\begin{align*}
    d_1 &= \sqrt{(x_1^2 - x^2) + (y_1^2 - y^2)} \\
    d_2 &= \sqrt{(x_2^2 - x^2) + (y_2^2 - y^2)} \\
    d_3 &= \sqrt{(x_3^2 - x^2) + (y_3^2 - y^2)}
\end{align*}
\]

3.2.4 AOA-Based Methods

An algorithm based on the mechanism of Angle of Arrival (AOA) detection has been proposed in (Zhang, Fan, Zhang, & Mo, 2010) which is named Trust Evaluation Base on AOA (TEBA). TEBA relies on the fact that a Sybil node can create multiple identities, but has only one physical location. Therefore, nodes whose signal phase differences are below a threshold are considered as Sybil nodes.
3.2.5 Distributed Methods

In (Li, Mittal, Caesar, & Borisov, 2012) Sybil control distributed algorithm has been proposed to take the control of the Sybil attack. It is an admission control mechanism for nodes in distributed systems in which nodes need to solve some computational puzzles periodically.

In (Ssu, Wang, & Chang, 2009), a distributed algorithm has been proposed for detecting the Sybil nodes with no need to have extra hardware or know about the number of neighbouring nodes. This algorithm does not need any central mechanism such as base stations or location-aware nodes.

3.2.6 Using Machine Learning Algorithms for Detecting Sybil Attack

(Zeng & Chen, 2010) Proposed a new protocol for WSN called SybilACO, which uses ant colony optimisation (ACO) algorithm to prevent Sybil attack. They considered the WSN as a social network with links between nodes as friendly relationships (see Figure 3-4).

![Social network model of Sybil attack](image)

Figure 3-4. Social network model of Sybil attack (Zeng & Chen, 2010)

The relations among trustful nodes have no problem. However, the relations among trustful nodes with Sybil nodes should be minimised. To minimise these relations, the proposed approach uses ACO. In this way, Sybil nodes become isolates. This is shown in Figure 3-5.
(Muraleedharan, Ye, & Osadciw, 2008) Used swarm intelligence algorithm to collect information about routes when the network is active to detect the Sybil nodes from their energy changes.

3.2.7 Authentication-Based Methods

(Butler, Ryu, Traynor, & McDaniel, 2009) Has proposed a new protocol which uses identifier-based encryption, and nodes are prohibited from acquiring identifiers. As a result, malicious nodes cannot acquire multiple identities and hence, there is no chance for Sybil attack.

(Dhamodharan & Vayanaperumal, 2015) Proposed CAM-PVM which is a message authentication algorithm. If a node is not authorised by the network or by the base station, the algorithm does not allow it to communicate with any other node in the network.

In (Amuthavalli & Bhuvaneswaran, 2014) a Random Password Generation (RPG) algorithm has been proposed that focuses on various traffic levels and security during data transmission in WSN. The RPG algorithm generates the routing table, which holds information about deployed nodes. The intermediate nodes in the route are identified between source and destination. The intermediate node’s information is compared with the RPG database during communication, and then the comparison results are used to decide whether these intermediate nodes are Sybil or normal.

In (Newsome., Shi, Song, & Perrig, 2004) several mechanisms were proposed for preventing Sybil attack which include:

- Detecting Sybil nodes using radio source test or randomised keys.
- Preventing Sybil attacks using registration of identifiers, code verification, and remote checking of code.

Radio source test technique requires each node to assign different signalling channels to its neighbours. This technique lacks efficiency because sensor nodes have many limitations. Identifier registration mechanism relies on voting and a central validation management unit in the network to identify Sybil nodes.

3.2.8 Detecting Sybil Attack in Mobile Wireless Sensor Networks

In (Banković, Fraga, Moya, & Vallejo, 2012) machine learning algorithms have been used to detect unknown attacks in wireless sensor networks by considering the attacks as an anomaly in network communication. In this work, the algorithm has been tested for Sybil attack on both static and mobile WSNs. The attacks have been treated as data outliers, which have been detected using clustering algorithms. The algorithms can achieve 100% detection rates when less than 52% of the nodes are malicious and can detect the presence of the attack if less than 80% of the nodes are malicious.

In (Sharmila & Umamaheswari, Detection of sybil attack in mobile wireless sensor networks, 2012) a clustering algorithm has been proposed to detect a Sybil node in a mobile WSN, which consists of three phases:

1. One of the nodes is considered as the base station. With the help of this node and after considering the packet drop rate, the nodes with minimum packet drop are chosen as cluster heads. Cluster heads consider the nodes with power value lower than a threshold as suspicious.
2. When neighbouring nodes send messages to the Sybil nodes collision will happen because all fake identifiers belong to the same physical node. Collisions can be used as an indication to detect the Sybil nodes.
3. Routing paths are checked to see if there any intermediate node (hub) between suspicious nodes. If yes, the nodes are not Sybil, otherwise, they are identified as Sybil nodes.

As can be seen, the algorithm consists of three complicated phases, which makes it unsuitable for MWSN with limited resources.

In (Sharmila & Umamaheswari, Node ID based detection of Sybil attack in mobile wireless sensor network, 2012) an algorithm has been proposed to detect the Sybil nodes in mobile WSN which requires nodes to register themselves to the base station. The base station validates and assigns an identifier to each legitimate node. Since this algorithm relies on the base station, it is not scalable.
In (Jamshidi, Zangeneh, Esnaashari, & Meybodi, 2017), an algorithm has been proposed to detect the Sybil attack in MWSN. The algorithm uses watchdog nodes to monitor the network and identify suspicious nodes. Watchdog nodes are normal nodes which are spread into the network and are specially programmed to collect information about the pattern of nodes movement. Therefore, the nodes are divided into two groups: sensor nodes (SN) and watchdog nodes (WN).

The algorithm consists of two phases: monitoring and detection. In the monitoring phase, when one SN moves in the network and lies in the neighbourhood of one WN, the WN stores some information about the SN. The information is stored in a data structure called “Moving_history”, which consists of two columns—Node_ID and Bit_label—where the former shows the node identity, and the latter contains the binary code of each WN which has previously had this node in its neighbourhood. To minimise the size of Moving_history, the minimum number of bits are considered to uniquely identify each WN. If the number of WNs is q, the binary code of each WN consists log₂ (q) bits. Figure 3-6 shows an example of deploying nodes in the network which consists of three watchdog nodes: W1, W2, and W3, 10 Sybil nodes: S1-S10, and several sensor nodes including a, b, c, d, x, y, z, u, and v. Figure 3-7 shows the Moving_history of W1 after the first round of the algorithm.

![Figure 3-6. An example of node locations in WSN Jamshidi et al., (2017).](image-url)
<table>
<thead>
<tr>
<th>Node_ID</th>
<th>Bit_label</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1-S10</td>
<td>00</td>
</tr>
<tr>
<td>x</td>
<td>00</td>
</tr>
<tr>
<td>y</td>
<td>00</td>
</tr>
<tr>
<td>z</td>
<td>00</td>
</tr>
<tr>
<td>u</td>
<td>00</td>
</tr>
<tr>
<td>v</td>
<td>00</td>
</tr>
<tr>
<td>d</td>
<td>00</td>
</tr>
</tbody>
</table>

Figure 3-7. The moving history of W1 in Figure 3-7 (Jamshidi, Zangeneh, Esnaashari, & Meybodi, 2017).

The algorithm relies on this fact when a malicious node moves in the network; all associated Sybil identities are also moved with it. Figure 3-8 shows the location of nodes after the second round of the algorithm when nodes relocated in the network. At the end of each round, WNs communicate with each other to send the history of nodes which were previously located in their neighbourhood to the WN which has these nodes in its neighbourhood after the current round of the algorithm. The new Moving_history of W1 and W3 are shown in Figure 3-9. As it can be seen in the figure, nodes x, y, and S1-10 moved to the neighbourhood of W3. Consequently, their histories are also moved from W1 to W3. The binary code of W3 (10) is also concatenated to their previous Bit_label, which consists the binary code of W1 (00).

Figure 3-8. Location of nodes after the second round of the algorithm (Jamshidi, Zangeneh, Esnaashari, & Meybodi, 2017).
In this way, Sybil nodes have the same Bit_label. Therefore, in the detection phase, each WN searches its Moving_history to find the same Bit_labels with the length greater than a threshold. These nodes are considered as Sybil.

In (Shehni, Faez, Farshad, & Kelarestaghi, 2017) an algorithm has been proposed for detecting the Sybil attack in MWSN using watchdog nodes. In this algorithm, each watchdog uses two data structures, which are upper triangular matrices whose number of rows and columns equals to the number of nodes (see Figure 3-10). Each element \((i,j)\) in \(A^k_{\text{co-prs}}\) is determined as follows:

- \((1,1)\): If both nodes \(i\) and \(j\) are in the neighbourhood of the watchdog node
- \((0,0)\): If none of the nodes \(i\) or \(j\) is in the neighbourhood of the watchdog node
- \((1,0)\): If node \(i\) is the neighbour of the watchdog node but node \(j\) not.
- \((0,1)\): If node \(j\) is the neighbour of the watchdog node but node \(i\) not.

After each round of the algorithm, \(A^k_{\text{co-prs}}\) is updated and renamed as \(A^{k+1}_{\text{co-prs}}\).

![Figure 3-9](image1.png)

**Figure 3-9.** Left: Moving_history of \(W1\); Right: Moving_history of \(W3\) after the second round (Jamshidi, Zangeneh, Esnaashari, & Meybodi, 2017).

![Figure 3-10](image2.png)

**Figure 3-10** Data structures used in (Shehni, Faez, Farshad, & Kelarestaghi, 2017):

- **a)** \(A^k_{\text{co-prs}}\)
- **b)** \(C^k_{\text{co-prs}}\)
In each round of the algorithm, the state diagram in Figure 3-11 is used to calculate each element of $C^k_{co-prs}$ from $A^{k-1}_{co-prs}$ and $A^k_{co-prs}$.

![State Diagram](image)

**Figure 3-11. The state diagram for calculating elements of $C^k_{co-prs}$ (Shehni, Faez, Farshad, & Kelarestaghi, 2017)**

After several rounds of the algorithm, all watchdog nodes send their $C^k_{co-prs}$ to a designated watchdog to aggregate the elements. In the end, if the element $(i, j)$ in $C^k_{co-prs}$ exceeds a threshold, it means that $i$ and $j$ are copies of the same node (i.e., Sybil identities). In simple words, the algorithm counts on this fact that the pattern of movement for Sybil nodes is the same. So, if they are both present or absent in the neighbourhood of a watchdog node, they are moving together.

The main issues of this algorithm are as follows:

- The memory overhead is high because of storing large matrices in watchdog nodes.
- Since a watchdog must process all $C^k_{co-prs}$, it will become a single point of failure. If this node becomes unavailable (for example, because of running out of the battery) the algorithm will crash.
- Since one watchdog node must process all $C^k_{co-prs}$ matrices, there are some serious concerns on the scalability of the algorithm.
3.3 Summary

In this chapter, I reviewed some of the existing researches about detecting the Sybil attack. As it was discussed in the previous sections, there have been many pieces of research around detecting and preventing Sybil attack in static WSN most of which are not suitable for mobile wireless sensor networks (MWSN) because most of these algorithms rely on the position of the nodes, RSSI, or collaboration between neighbouring nodes. (Yu, Lu, & Kuo, 2008) compared the complexity of some of the researches discussed in this chapter. The results are depicted in Table 3-1 where n is the number of nodes, d is the average number of neighbours for each node, and g is the number of sent messages from each node. As it can be seen from the table, these methods suffer from a significant overhead (both memory and communication) which could be a barrier to their scalability.

Table 3-1. Comparing the complexity of several Sybil detection algorithms (Yu, Lu, & Kuo, 2008)

<table>
<thead>
<tr>
<th>Research</th>
<th>Memory</th>
<th>Communication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Broadcast (Parno, Perrig, &amp; Gligor, 2005)</td>
<td>O(d×n)</td>
<td>O(n^2)</td>
</tr>
<tr>
<td>Deterministic Multicast (Parno, Perrig, &amp; Gligor, 2005)</td>
<td>O(g×n)</td>
<td>O(\frac{g \ln g \sqrt{n}}{d})</td>
</tr>
<tr>
<td>Randomised Multicast (Parno, Perrig, &amp; Gligor, 2005)</td>
<td>O(\sqrt{n}×n)</td>
<td>O(n)</td>
</tr>
<tr>
<td>Line-Selected Multicast (Parno, Perrig, &amp; Gligor, 2005)</td>
<td>O(\sqrt{n}×n)</td>
<td>O(n√n)</td>
</tr>
<tr>
<td>(Brooks, Govindaraj, Pirretti, Vijaykrishnan, &amp; Kandemir, 2007)</td>
<td>N.A.</td>
<td>O(n log n)</td>
</tr>
<tr>
<td>(Zhang, Khanapure, Chen, &amp; Xiao, 2009)</td>
<td>O(\sqrt{n}×n)</td>
<td>O(n√n)</td>
</tr>
<tr>
<td>(Li &amp; Gong, 2009)</td>
<td>O(d×n)</td>
<td>O(d×n√n)</td>
</tr>
</tbody>
</table>

In (Andalib & Jamshidi, 2016), the detection rate of some of the previous methods has been reported as shown in Figure 3-12 and Figure 3-13 which the former depicts true detection rate and the latter depicts false detection rate. (Demirbas & Song, 2006) has a very high true detection rate which means it can detect all Sybil nodes correctly. However, its false detection rate is also high which means it mistakenly considers some normal nodes as Sybil. On the other hand, (Dhamodharan & Vayanaperumal, 2015) and (Amuthavalli & Bhuvaneswaran, 2014) have very low false detection rates, but their true detection rates are also low. Therefore, none of these algorithms could achieve high true detection rate and low false detection rate at the same time. After all, these algorithms are only suitable for static WSN.
Figure 3-12. The true detection rate of several Sybil detection algorithms

Figure 3-13. The false detection rate of several Sybil detection algorithms

I also explained several algorithms for detection of Sybil attack in MWSN. A clustering algorithm proposed by (Sharmila & Umamaheswari, Detection of sybil attack in mobile wireless sensor networks, 2012) in which collision is an indication for detecting the Sybil nodes. This algorithm suffers from a high complexity.

Sharmila and Umamaheswari also proposed another algorithm for detecting the Sybil attack in MWSN, in which nodes must be validated by the base station (Sharmila & Umamaheswari, Node ID based detection of Sybil attack in mobile wireless sensor network, 2012). This algorithm is centralised and is not scalable.
I also explained the algorithm proposed by (Jamshidi, Zangeneh, Esnaashari, & Meybodi, 2017) and (Shehni, Faez, Farshad, & Kelarestaghi, 2017) for detecting mobile Sybil nodes in MWSN which uses watchdog nodes. Watchdog nodes are normal trustful nodes in the WSN which have been initially programmed to observe the network and report malicious behaviours. The former algorithm suffers from three problems:

- Its convergence is very slow which means that the algorithm needs many rounds of execution until its detection rate becomes acceptable.
- Observer nodes do not act independently when detecting the Sybil nodes because they need to send special messages to each other to detect the Sybil nodes.

While the shortcomings of the latter are as follows:

- High memory usage
- Single point of failure
- Lack of scalability

In the next chapter, I propose a new algorithm for detecting the Sybil attack in MWSN which overcomes the shortcomings of the algorithms proposed by (Jamshidi, Zangeneh, Esnaashari, & Meybodi, 2017) and (Shehni, Faez, Farshad, & Kelarestaghi, 2017).
Chapter Four: Methodology

As it was discussed in the previous chapters, Sybil is one of the most serious attacks in wireless sensor network (WSN). So, it is important to detect it. Many researchers have been conducted on detecting the Sybil attack in WSN. However, most of them rely on the location of sensor nodes and their signal strengths, which means these algorithms are not suitable for mobile wireless sensor networks (MWSN). Of course, some algorithms have been proposed for detecting the Sybil attack in MWSN, but they suffer from slow convergence and scalability.

Therefore, I decided to propose a new algorithm for detecting the Sybil nodes in mobile wireless which addresses the shortcomings of the existing algorithms. The proposed algorithm uses observer nodes to keep the number of times a node has appeared as a common neighbour between itself and its neighbours. After collecting some information about its neighbours, each observer node considers the nodes which appeared more than a threshold as critical, and the nodes having all of them in their neighbourhood will be considered suspicious nodes. In this chapter, the research methodology and the proposed algorithm are detailed.

4.1 Hypothesis

After experimentation, I will be able to answer various questions. This is the list of hypotheses that will be tested throughout this research project:

- The cooperation among observer nodes makes the proposed algorithm distributed and suitable for MWSN.
- Using observer nodes results in high accuracy for detecting the Sybil nodes.
- Memory, computation, and communication overheads of the proposed algorithm are low to make it suitable for WSN.

4.2 Research Approach

A major factor in the success of a research is choosing a suitable research approach which itself depends on the research problem. According to (Creswell & Creswell, 2018), there are three types of research approaches as follows:

- Qualitative research:
  - An approach for understanding the meaning individuals and groups ascribe to a human or social problem
  - Emerging questions
- Flexible written report
- Analysis building from particular data to general themes (inductive)

- **Quantitative research:**
  - An approach for testing objective theories by examining the relationships among variables (deductive)
  - Numbered data which can be analysed using statistical procedures
  - Structured written report

- **Mixed methods research:**
  - Collection of both qualitative and quantitative data
  - Integrating the two forms of data
  - May involve both philosophical assumptions and theoretical frameworks
  - Assumes a more complete understanding of a research problem than using one of the approaches alone

The research method chosen for this thesis is simulation because it utilizes testing the scenarios that are difficult or expensive to experiment with in the real world. It's particularly useful to test networking attacks and their mitigation.

Since this research is based on simulation and the collected data is numeric, the methodology used for this research is quantitated which is detailed in this chapter.

### 4.3 System Assumptions and Attack Model

I divide nodes in the sensor network into two sets: ordinary sensor nodes (SN) and observer nodes (ON), which are randomly distributed in a two-dimensional area. SNs perform normal operations such as data gathering and sending data while WNs are responsible for detecting the Sybil attack. Each node has its own unique ID and is not necessarily aware of its location. All nodes are mobile and move in the network area according to a mobility model such as random waypoint. The nodes communicate with each other via a wireless radio channel and broadcast information in an Omni-directional mode.

According to (Newsome, Shi, Song, & Perrig, 2004) there are several categories of the Sybil attack of which I have chosen direct, fabricated and simultaneous Sybil attack for this research. The nodes in the network are divided into two groups: legitimate nodes and malicious nodes of which the latter cheat their neighbours by creating multiple identities (Sybil nodes). Any communication with a Sybil node is handled by its corresponding malicious node. Malicious nodes try to trick the legitimate nodes to make them believe that they have many neighbours. Since Sybil nodes do not really exist, they may affect many of the network protocols
and voting algorithms. I assume malicious nodes (and their associated Sybil nodes) move in the network similarly to legitimate nodes.

Finally, it is assumed that the adversary tries to compromise some legitimate nodes and reprogram them to make them malicious. However, observer nodes cannot be captured and programmed by the adversary.

4.4 Proposed Algorithm
In the Sybil attack, malicious node illegitimately forges multiple (fake) identities. This means that the malicious node replicates itself to make many copies to confuse and collapse the network. For example, in Figure 4-1, the malicious node forges four different IDs. Its neighbouring nodes think they have four different nodes in their vicinity.

![Figure 4-1. A malicious node which forges four IDs](image)

In the proposed algorithm, observer nodes are used to identify Sybil nodes in mobile wireless networks. Observer nodes are normal trustful nodes in the WSN which have been programmed initially to observe the network and report malicious behaviours. The proposed algorithm will consist of two phases:

- **Observation:** After setting up the network, all observer nodes broadcast hello messages to identify their neighbours and construct routing tables. An observer node, after identifying its neighbours, will get the list of common neighbours with each neighbour and stores them in its memory which is called the history. Then, the observer keeps the number of times a node has appeared as a common neighbour between itself and its neighbours. During the time, observer nodes continuously update their history.

- **Detection:** After collecting some information about its neighbours and counting the number of times each node appears in the list of common neighbours, each observer node considers the nodes whose counters are above a threshold $\theta$ (which is determined through
experimentation) as critical. The nodes having all critical nodes in their neighbourhood will be considered as suspicious nodes. For example, in the network in Figure 4-2, nodes 1-6 are normal, node M and nodes 7-14 are Sybil nodes. Node V is the observer node.

The algorithm identifies nodes 4, 5, and 6 as critical nodes because their counters are above the threshold. This means that these nodes are probably adjacent to Sybil nodes. Thus, the nodes having all of them into their neighbouring list become suspicious: 2, 7-14, and M.

Note that there is a chance for normal nodes to be mistakenly considered suspicious (in the above example node 2 is normal but it is considered suspicious). To reduce this chance, observer nodes will communicate with each other and exchange their suspicious list to identify the common nodes. These nodes are announced as malicious nodes.

<table>
<thead>
<tr>
<th>ID</th>
<th>Counter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>13</td>
</tr>
<tr>
<td>5</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>11</td>
<td>8</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
</tr>
<tr>
<td>13</td>
<td>8</td>
</tr>
<tr>
<td>14</td>
<td>8</td>
</tr>
<tr>
<td>M</td>
<td>3</td>
</tr>
</tbody>
</table>

Sybil Nodes: 7-14

<table>
<thead>
<tr>
<th>ID</th>
<th>Counter</th>
<th>ID</th>
<th>Counter</th>
<th>CNB(_i,v)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
<td>13</td>
<td>{2,4,7,9,13,14}</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>{1,3,4,5,6,8,10,11,12}</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>{2,5,6,7,10,13}</td>
</tr>
<tr>
<td>4</td>
<td>13</td>
<td>4</td>
<td>4</td>
<td>{1,2,5,6,8,11,14,M}</td>
</tr>
<tr>
<td>5</td>
<td>13</td>
<td>5</td>
<td>5</td>
<td>{2,3,4,6,10,11,13,M}</td>
</tr>
<tr>
<td>6</td>
<td>12</td>
<td>6</td>
<td>6</td>
<td>{2,3,4,5,7,8,9,13,15,M}</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>{4,5,6,8,9,11,14}</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>{4,5,6,7,9,12,13}</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>9</td>
<td>9</td>
<td>{4,5,6,7,8}</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>{4,5,6,7,8,11,12,13,14}</td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td>11</td>
<td>11</td>
<td>{4,5,6,10,13}</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>12</td>
<td>12</td>
<td>{4,5,6,7,9,11,14}</td>
</tr>
<tr>
<td>13</td>
<td>8</td>
<td>13</td>
<td>13</td>
<td>{4,5,6,7,8,9,14}</td>
</tr>
<tr>
<td>14</td>
<td>8</td>
<td>14</td>
<td>14</td>
<td>{4,5,6,9,10,11,13}</td>
</tr>
<tr>
<td>M</td>
<td>3</td>
<td></td>
<td></td>
<td>{4,5,6,7,10,14}</td>
</tr>
</tbody>
</table>

\(\theta = 10.5\)

Figure 4-2. The idea of the proposed algorithm

32
4.5 Simulation Setup

To implement the proposed algorithm, to achieve more flexibility, instead of using existing simulators, I wrote a C++ program for simulating the wireless sensor network based on my assumptions and the model discussed in Section 4.2. Implementing the simulator with C++ offered higher flexibility to me than using MATLAB or other existing simulators. Before starting the implementation phase, I investigated several network simulators, more particularly NS2, OMNET++ and J-Sim. I realized they are too complicated and they lack enough support which might cause problems in the middle of my simulation. I read several papers and theses which compared the WSN simulators in terms of performance, extensibility, usability, and platforms (Turower, 2016), (Gupta, Ghonge, Thakare, & Jawandhiya, 2016) and (Chen, J. Branch, Zhu, & Szymanski, 2005). Since the extensibility was a key factor for me to implement my algorithm, the model of attack and measure the performance factors I needed for the evaluation, I decided not to take the risk to spend lots of time learning and using one of these simulators and then stuck in the middle of so many problems for implementation. I implemented my simulator which is inspired by some of the existing simulators more particularly SENSE (https://www.ita.cs.rpi.edu/) which is an open source simulator with rich documentation.

I validated my simulator by implementing the same scenarios explained in (Jamshidi, Zangeneh, Esnaashari, & Meybodi, 2017) and (Shehni, Faez, Farshad, & Kelarestaghi, 2017) and compared the results to ensure they match with the results reported in the two papers. Having the code of my own simulator I can extend my experimentation in the future, revising my algorithm and using different protocols and network models.

The following parameters have been defined for network configuration:

- Number of rounds
- Number of nodes
- Number of observer nodes
- Size of the network field
- Number of malicious and captured nodes
- Transmission range
- Number of Sybil nodes generated by each malicious node
Threshold

As it will be discussed in the next chapter, these parameters have been set appropriately for each experimentation. I also implemented node structures and their deployment in the network area and their movement.

For the physical layer, Two Ray Ground (TRG) model has been used. Instead of assuming there is a direct line of sight between nodes, TRG considers not only a direct path but also a ground reflection path (see Figure 4-3). The following formula calculates the received power at distance $d$ (Eenennaam, 2008):

$$ P_r(d) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4} $$

(4-1)

Where $h_r$ and $h_t$ are the heights of receive and transmit antennas, respectively. TRG shows a faster power loss than other models as the distance increases.

![Figure 4-3. Two Ray Ground Propagation (Eenennaam, 2008)](image)

The routing protocol for the simulation is Ad hoc On-Demand Distance Vector (AODV) (Perkins, Belding-Royer, & Das, 2003) which is one of the reactive protocols for WSN. In this protocol, each node has its own routing table. When node A wants to send a message to node B, it checks its routing table first. If a route to B is found, the message will be sent. Otherwise, node A initiates a route request process and broadcasts a route request packet (RREQ). Any node that receives the RREQ packet searches its routing table for a route to B. If found, a route reply (RREP) packet is sent to A with information about the route to B. Otherwise, the RREQ packet is forwarded to other nodes.
In the simulator, malicious nodes are programmed to capture nodes and forge Sybil identities. The most important part of the implementation has been programming observer nodes to implement the proposed algorithm. These nodes are monitoring the network and communicate with each other to detect suspicious nodes.

The simulator architecture is depicted in Figure 4-4. A typical wireless sensor node is composed of several modules (e.g., timer, CPU, radio, network layers) each of which is based on one or more tasks. The proposed simulator defines the classes corresponding to each module, for example, routing, application, Media Access Control (MAC) and physical layer. These modules have been implemented as C++ classes derived from a core simulator layer class. Tasks have been implemented as methods within a module class.

The network class encompasses several other classes as follows:

- **Energy** class is used to represent the level of energy and energy consumption in a mobile node.
- **Battery** class is used to represent the sensor nodes power supply.
- **Propagation** class is used to predict the received signal power of each packet.
- **Delay** class is used to compute the time a packet needs to traverse a link.
- **Malicious** class that is used to implement the behaviour of malicious nodes.
- **Packet** class is used to store information of the packets transmitted between different nodes in the network.
- **Event** class is used to provide basic functionality for all events.
- **Data generator** class is used to generate sensed data in the network.
- **Error** class is used to handle packet losses.

### 4.6 Data Collection Process

This research involves two types of data collection processes as follows:

- Literature review process: Similarly to other researches, the literature review is one major source of data for this research. Published researches have been collected from academic databases (e.g., IEEEXplore, ACM, etc.), books, and websites.
- Experimental data gathering process: The second major source of data is the data result from the simulation. A network simulator is set up for implementing the proposed algorithm and several metrics are measured to evaluate the proposed algorithm. These metrics are explained in the next chapter.

### 4.7 Summary

In this chapter, I proposed a new algorithm for detecting the Sybil attack in mobile wireless sensor networks. The proposed algorithm uses observer nodes to monitor the network and find suspicious nodes. Observer nodes are normal nodes which are specially programmed to communicate with other nodes and get the list of their neighbours. The algorithm is based on this assumption that when a malicious node moves in the network all its associated Sybil ID’s also move with it. When there are many common neighbours between an observer node and other nodes, it means that they are adjacent to a malicious node which forges Sybil IDs. Therefore, the nodes which appeared in many neighbouring lists are identified as critical and the nodes having all these critical nodes in their neighbouring lists are considered as suspicious. In this way, some legitimate nodes may be considered as suspicious too. To reduce the chance of mistakenly considering legitimate nodes as suspicious, the observer nodes communicate with each other to refine the list of suspicious nodes.

The algorithm has been simulated in C++. In the next chapter, I evaluate the proposed algorithm and compare it with similar algorithms.
Chapter Five: Evaluation of the Proposed Algorithm

In the previous chapter, I detailed the proposed algorithm which uses observer nodes to detect the Sybil attack in mobile wireless networks (MWSN). To implement the proposed algorithm I provided a simulator in C++. In this chapter, I evaluate the proposed algorithm and compare it with similar researches.

5.1 Data Collection Process

To simulate the network and compare the performance of the proposed algorithm with other researches, I reviewed the literature and configured the network as follows:

Table 5-1. Network configuration for simulation.

<table>
<thead>
<tr>
<th>Area</th>
<th>100x100 m²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes (N)</td>
<td>300</td>
</tr>
<tr>
<td>Malicious nodes (M)</td>
<td>5</td>
</tr>
<tr>
<td>Sybil identifiers forged by each malicious node (S)</td>
<td>10</td>
</tr>
<tr>
<td>Observer nodes (ON)</td>
<td>4</td>
</tr>
<tr>
<td>Transmission range</td>
<td>10 m</td>
</tr>
<tr>
<td>Threshold (θ)</td>
<td>0.6 x d</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Random Waypoint</td>
</tr>
</tbody>
</table>

There are N – (M+ON) normal nodes in the network. In the beginning, nodes are distributed normally in the simulation area. In the table, θ is considered as 0.6 x d, where d is the number of neighbours of observer nodes. Of course, as it will be discussed later, to evaluate several aspects of the proposed algorithm, I change some of the network parameters for each experimentation.

5.2 Evaluation metrics

The proposed algorithm has been evaluated (and compared with similar researches) using the following metrics:

**Overhead:**

- **Memory:** The amount of memory required by observer nodes to store information about repeating nodes in their neighbourhood.
- **Communication**: The amount of communication required by observer nodes to get the information about neighbouring nodes and exchanging information with other observer nodes.
- **Computation**: Time complexity for running the algorithm by observer nodes.

**Detection Rate:**

- **True detection rate**: How many attackers have been correctly identified as attackers?
- **False detection rate**: How many normal nodes have been mistakenly identified as attackers?

5.3 The Complexity of the Proposed Algorithm

5.3.1 Memory Overhead

Since the proposed algorithm is performed by observer nodes (ON), only ONs suffer from memory overhead. Each ON needs to allocate part of its memory to store common neighbours with its own neighbours (history). If each node has $d$ neighbours on average, the imposed memory overhead is $O(ON \times d)$. In addition, each ON keeps a counter for each node to measure the number of times the node appears in the history. If there are $n$ nodes in the network, the memory used for counters is $O(ON \times n)$. Therefore, in total, the memory overhead in the proposed algorithm is $O(ON \times d + ON \times n)$. Since $d \ll n$, the memory overhead equals to $O(ON \times n)$.

5.3.2 Communication Overhead

Energy consumption is a big concern in WSN because nodes are not rechargeable. The most energy consuming operation for sensor nodes is sending packets. Therefore, the number of transmitted packets in the network during the execution of the algorithm is one of our major concerns. When sensor nodes move to a new location, they broadcast “Hello” message to update their neighbouring table or to request a new route. Since, this is one of the requirements for mobile sensor networks, “Hello” messages sent by nodes are not considered as overhead for the proposed algorithm. At the end of each round, each ON sends (broadcasts) a message to its neighbours to receive their neighbours. So, the communication overhead is $O(ON)$. In addition, ONs communicate with each other to exchange the list of suspicious nodes. This happens at the end of each round. So, the imposed communication overhead for exchanging suspicious nodes is also $O(ON)$. Therefore, the communication overhead of the proposed algorithm in each round is $O(ON)$. In total, after $p$ rounds, the communication overhead of the proposed algorithm is $O(ON \times p)$. 

38
5.3.3 Computation Overhead

Since only ONs are supposed to perform the algorithm, the computation overhead is considered for ONs only. Each ON traverses the history to calculate the frequency of each node in the list. If each entry in the history consists $k$ IDs on the average, to calculate the frequency of the nodes, the computation complexity in each round is $O(ON \times d \times k)$. Then, the list of node frequencies should be traversed to identify suspicious nodes. The complexity of this is $O(ON \times N)$. Then, ONs exchange their suspicious list and find the intersection. If we assume on average there are $S$ suspicious nodes for each ON, the complexity of finding the intersection is $O(ON \times S)$. Therefore, the total computation overhead is $O(ON \times d \times k + ON \times N + ON \times S)$. Since $S << d \times k << n$ the computation overhead of the proposed algorithm is $O(ON \times d \times k)$.

Table 5-2 compares the overhead of the proposed algorithm with (Jamshidi, Zangeneh, Esnaashari, & Meybodi, 2017) and (Shehni, Faez, Farshad, & Kelarestaghi, 2017).

Table 5-2. Comparing the complexity of the proposed algorithm with (Jamshidi, Zangeneh, Esnaashari, & Meybodi, 2017)

<table>
<thead>
<tr>
<th>Research</th>
<th>Memory</th>
<th>Communication</th>
<th>Computation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Jamshidi, Zangeneh, Esnaashari,</td>
<td>$O(ON \times \log n \times p)$</td>
<td>$O(ON \times p)$</td>
<td>$O(n \times d \times p)$</td>
</tr>
<tr>
<td>&amp; Meybodi, 2017)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Shehni, Faez, Farshad, &amp; Kelarestaghi, 2017)</td>
<td>$O(ON \times n^2)$</td>
<td>$O(ON \times n^2)$</td>
<td>$O(n^2 \times p)$</td>
</tr>
<tr>
<td>Proposed Algorithm</td>
<td>$O(ON \times n)$</td>
<td>$O(ON \times p)$</td>
<td>$O(n \times ON \times p)$</td>
</tr>
</tbody>
</table>

In Table 5-2, $n$ is the number of nodes, $d$ is the average number of neighbours for each node, and $p$ is the number of rounds. As it can be seen from the table, the complexity of (Shehni, Faez, Farshad, & Kelarestaghi, 2017) is higher than the other two algorithms. The proposed algorithm and (Jamshidi, Zangeneh, Esnaashari, & Meybodi, 2017) have the same communication complexity. However, since in their algorithm the moving history of the nodes is stored as a sequence which grows after each round, the memory overhead of Jamshidi et al., (2017) depends on the number of rounds and is arguably more than the proposed algorithm. Also, since $ON < d$ the computation overhead of my algorithm is better than (Jamshidi, Zangeneh, Esnaashari, & Meybodi, 2017). Therefore, the proposed algorithm has lower overhead than the algorithm proposed in (Jamshidi, Zangeneh, Esnaashari, & Meybodi, 2017). In addition, compared with algorithms discussed for static WSN in Chapter 3, the proposed algorithm has a lower overhead (see Table 3-1).
5.4 Detection Rates for Different Rounds of the Algorithm

Following the configuration in Table 5-1, I evaluated the algorithm in terms of detection rate for which I repeated the experiments 100 times and used the average of the results. Figure 5-1 shows the detection rate of the proposed algorithm in test rounds 50-300.

![Detection rate of the proposed algorithm in different rounds.](image)

As it can be seen from the figure, as the time passes and observer nodes communicate with other nodes to collect more information about their neighbours, they can detect the Sybil nodes more accurately.

In addition, the false detection rate is initially high which means some of the normal nodes are detected as Sybil nodes. However, after several rounds, observer nodes know more about their neighbours and communicate with each other to exchange the list of suspicious nodes. As a result, the algorithm becomes more accurate in detecting suspicious nodes.

5.5 Impact of Various Factors on Detection Rates

To fully investigate the performance of the proposed algorithm, I conducted several experiments to evaluate the impact of several parameters on the detection rate. The experiments are detailed in this section.

5.5.1 Number of Sybil Identifiers Forged by Malicious Nodes

In this experiment, I studied the impact of the number of Sybil identifiers forged by a malicious node (S) on both true and false detection rates. The configuration of the network is shown in Table 5-3. Parameter S takes the values 5, 10, 15, and 20.
Table 5-3. Network configuration for investigating the impact of the number of Sybil IDs

<table>
<thead>
<tr>
<th>Area</th>
<th>100x100 m²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes (N)</td>
<td>300</td>
</tr>
<tr>
<td>Malicious nodes (M)</td>
<td>5</td>
</tr>
<tr>
<td>Sybil identifiers forged by each malicious node (S)</td>
<td>5, 10, 15, 20</td>
</tr>
<tr>
<td>Observer nodes (ON)</td>
<td>4</td>
</tr>
<tr>
<td>Transmission range</td>
<td>10 m</td>
</tr>
<tr>
<td>Threshold (θ)</td>
<td>0.6 x d</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Random Waypoint</td>
</tr>
</tbody>
</table>

Figure 5-2 shows the results for true detection rate. As it can be seen from the figure, the lowest detection rate is achieved for S = 5. This is because spreading fewer Sybil identifiers causes the number of identifiers in the Sybil group to reach below the threshold θ and so the total Sybil group will not be detected.

![True Detection Rate](image)

**Figure 5-2. The true detection rate for various number of Sybil IDs**

Figure 5-3 shows the false detection rate for different values of S. As shown in the figure, parameter S does not have any significant impact on the false detection rate because all Sybil identifiers belong to a single malicious node, so different values of this parameter do not affect the probability of a normal node to join a Sybil group.
5.5.2 Number of Malicious Nodes

In this experiment, we studied the effect of the number of malicious nodes (M) on the detection rate of the proposed algorithm. The configuration of the network is shown in Table 5-4. Parameter M takes the values 2, 5, 8, and 10.

Table 5-4. Network configuration for investigating the impact of the number of malicious nodes

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>100x100 m²</td>
</tr>
<tr>
<td>Nodes (N)</td>
<td>300</td>
</tr>
<tr>
<td>Malicious nodes (M)</td>
<td>2, 5, 8, 10</td>
</tr>
<tr>
<td>Sybil identifiers forged by each malicious node (S)</td>
<td>10</td>
</tr>
<tr>
<td>Observer nodes (ON)</td>
<td>4</td>
</tr>
<tr>
<td>Transmission range</td>
<td>10 m</td>
</tr>
<tr>
<td>Threshold (θ)</td>
<td>0.6 x d</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Random Waypoint</td>
</tr>
</tbody>
</table>

Figure 5-4 shows the true detection rate for different values of M. As it can be seen from the figure, M does not have a significant impact on true detection rate in the proposed algorithm. That is because the proposed algorithm takes the node movement into account and each malicious node moves with its own Sybil identities independently.
Figure 5-4. The true detection rate for various numbers of malicious nodes

Figure 5-5 shows the impact of $M$ on the false detection rate. As expected, $M$ affects false detection rate. Increasing $M$ will increase false detection rate, because the existence of so many malicious nodes increases the probability of a certain normal node to move with a malicious node during the initial several rounds. It is noted that after 250 rounds, the false detection rate becomes negligible regardless of the value of $M$.

Figure 5-5. The false detection rate for various numbers of malicious nodes
5.5.3 Number of Observer Nodes

In this experiment, I examined the impact of the number of observer nodes (ON) on the accuracy of the proposed algorithm. The configuration of the network is shown in Table 5-5. Parameter ON takes the values 2, 4, 6, 8, and 10.

Table 5-5. Network configuration for investigating the impact of the number of observer nodes

<table>
<thead>
<tr>
<th>Area</th>
<th>100x100 m²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes (N)</td>
<td>300</td>
</tr>
<tr>
<td>Malicious nodes (M)</td>
<td>5</td>
</tr>
<tr>
<td>Sybil identifiers forged by each malicious node (S)</td>
<td>10</td>
</tr>
<tr>
<td>Observer nodes (ON)</td>
<td>2, 4, 6, 8, 10</td>
</tr>
<tr>
<td>Transmission range</td>
<td>10 m</td>
</tr>
<tr>
<td>Threshold (θ)</td>
<td>0.6 x d</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Random Waypoint</td>
</tr>
</tbody>
</table>

Obviously, using more observer nodes improves the accuracy of the algorithm. However, the ideal situation would be using a minimal number of observer nodes to accurately detect the Sybil nodes. Using many observer nodes will increase the overhead of the algorithm. As it can be seen from the figure, having 2 observer nodes results in a low true detection rate. However, using 4 observer nodes significantly improves the true detection rate. Using more than 4 observer nodes does not result in significantly better detection rate. Thus, it seems that the optimal number of observer nodes for this network configuration is 4.

![Figure 5-6. The true detection rate for various number of observer nodes](image-url)
Figure 5-7 shows the false detection rate for different values of ON. Similarly to the true detection rate, the optimal value of ON is 4.

![False Detection Rate](image)

**Figure 5-7. The false detection rate for various number of observer nodes**

To determine the optimal number of observer nodes for the designated network, I conducted a couple of more experiments as detailed in the following. In the first experiment, I investigated the impact of the number of observer nodes and the number of malicious nodes on the accuracy of the proposed algorithm. Table 5-6 shows the configuration of the network.

**Table 5-6. Network configuration for investigating the impact of the number of observer nodes and malicious nodes**

<table>
<thead>
<tr>
<th>Area</th>
<th>100x100 m²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes (N)</td>
<td>300</td>
</tr>
<tr>
<td>Malicious nodes (M)</td>
<td>2, 5, 8, 10</td>
</tr>
<tr>
<td>Sybil identifiers forged by each malicious node (S)</td>
<td>10</td>
</tr>
<tr>
<td>Observer nodes (ON)</td>
<td>2, 4, 6, 8, 10</td>
</tr>
<tr>
<td>Transmission range</td>
<td>10 m</td>
</tr>
<tr>
<td>Threshold ($\theta$)</td>
<td>0.6 x d</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Random Waypoint</td>
</tr>
</tbody>
</table>

Figure 5-8 shows the impact of the number of observer nodes (ON) and the number of malicious nodes (M) on the true detection rate. The number of rounds for execution of the algorithm is 300. As it can be seen from the figure, when the number of malicious nodes increases,
true detection rate slightly decreases irrespective of the number of observer nodes. That is because the proposed algorithm takes the node movement into account and each malicious node moves with its own Sybil identities.

As per the figure, using more observer nodes increases true detection rate. However, the improvement is negligible for \( ON \geq 4 \), which means four observer nodes are enough for the network configuration considered in the experimentation. This optimal number of observer nodes may vary from network to network.

![True Detection Rate](image)

**Figure 5-8. The true detection rate for various numbers of observer nodes and malicious nodes**

Figure 5-9 shows the impact of the number of observer nodes and the number of malicious nodes on false detection rate. The number of rounds for execution of the algorithm is 300. As it can be seen from the figure, when the number of malicious nodes increases, false detection rate also increases irrespective of the number of observer nodes. This is because the existence of so many malicious nodes increases the probability of a certain normal node to move with a malicious node during the experimentation time.

As per the figure, using more observer nodes results in lower false detection rate. This is because more observer nodes can better cover the network area and exchange the list of suspicious nodes with each other and so, the chance of considering a normal node as Sybil decreases. Therefore, the optimal number of observer nodes for this experiment depends on the highest acceptable false detection rate (from the network designer’s point of view).
In the second experiment, I studied the impact of the number of observer nodes and the number of Sybil identities forged by a malicious node (S) on the detection rate of the proposed algorithm. The configuration of the network is shown in Table 5-7.

Table 5-7. Network configuration for investigating the impact of the number of observer nodes and Sybil IDs

<table>
<thead>
<tr>
<th>Area</th>
<th>100x100 m²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes (N)</td>
<td>300</td>
</tr>
<tr>
<td>Malicious nodes (M)</td>
<td>5</td>
</tr>
<tr>
<td>Sybil identifiers forged by each malicious node (S)</td>
<td>5, 10, 15, 20</td>
</tr>
<tr>
<td>Observer nodes (ON)</td>
<td>2, 4, 6, 8, 10</td>
</tr>
<tr>
<td>Transmission range</td>
<td>10 m</td>
</tr>
<tr>
<td>Threshold (θ)</td>
<td>0.6 x d</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Random Waypoint</td>
</tr>
</tbody>
</table>

Figure 5-10 shows that true detection rate increases when the number of observer nodes increases irrespective of the number of Sybil identifiers. For lower values of S, the true detection rate is lower because spreading fewer Sybil identities causes the number of identifiers in the Sybil group to reach below the threshold θ and so the total Sybil group will not be detected. It seems that the amount of improvement for ON ≥ 4 is negligible. Again, the results show that for the designated network configuration four observer nodes are enough.
Figure 5-10. The true detection rate for various numbers of observer nodes and Sybil IDs

As Figure 5-11 shows, increasing the number of observer nodes improves false detection rate irrespective of S. However, varying S does not have a major impact on the false detection rate. This is because all Sybil identifiers belong to a single physical malicious node. Thus, different values of S do not affect the probability of a normal node to join a Sybil group. However, as per the previous experiment, different values of M (number of malicious nodes) affect the probability of a normal node to join a Sybil group. Similarly to the true detection rate, the amount of improvement for ON ≥ 4 is negligible.

Figure 5-11. The false detection rate for various numbers of observer nodes and Sybil IDs
In conclusion, it seems that for the network configuration I considered for experimentations, four observer nodes should be enough to achieve good performance.

5.5.4 Threshold

In this experiment, I examined the impact of parameter $\theta$ (threshold). The values considered for $\theta$ are 0.4, 0.6, 0.8 times the number of neighbours ($d$). The configuration of the network is as per Table 5-8.

Table 5-8. Network configuration for investigating the impact of theta

<table>
<thead>
<tr>
<th>Area</th>
<th>$100 \times 100 \text{ m}^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes (N)</td>
<td>300</td>
</tr>
<tr>
<td>Malicious nodes (M)</td>
<td>5</td>
</tr>
<tr>
<td>Sybil identifiers forged by each malicious node (S)</td>
<td>10</td>
</tr>
<tr>
<td>Observer nodes (ON)</td>
<td>4</td>
</tr>
<tr>
<td>Transmission range</td>
<td>$10 \text{ m}$</td>
</tr>
<tr>
<td>Threshold ($\theta$)</td>
<td>$(0.4, 0.6, 0.8) \times d$</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Random Waypoint</td>
</tr>
</tbody>
</table>

Figure 5-12 shows the impact of $\theta$ on true detection rate. As it can be seen from the figure, lower values for $\theta$ results in better true detection rate. This is because some of the Sybil identities may be missed by their neighbours when sending “Hello” messages. This usually happens due to collisions. As a result, the frequency of such IDs may fall below the threshold and cannot be detected by the algorithm. Using a lower value for $\theta$ decreases the chance of missing these IDs in the algorithm.
Figure 5-13 shows the false detection rate for different values of $\theta$. As it is evident in the figure, low values for $\theta$ result in high false detection rate because even normal nodes with low frequency will be considered suspicious.

![False Detection Rate](image)

Figure 5-13. The false detection rate for different values of theta

5.5.5 Transmission Range

In this experiment, I studied the impact of transmission range on the detection rate of the proposed algorithm. The configuration of the network is shown in Table 5-9. Transmission range is from 1 to 20 meters. The number of rounds for execution of the algorithm is 300.

<table>
<thead>
<tr>
<th>Area</th>
<th>$100\times100$ m$^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes (N)</td>
<td>300</td>
</tr>
<tr>
<td>Malicious nodes (M)</td>
<td>5</td>
</tr>
<tr>
<td>Sybil identifiers forged by each malicious node (S)</td>
<td>10</td>
</tr>
<tr>
<td>Observer nodes (ON)</td>
<td>4</td>
</tr>
<tr>
<td>Transmission range</td>
<td>1-20 m</td>
</tr>
<tr>
<td>Threshold ($\theta$)</td>
<td>$0.6 \times d$</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Random Waypoint</td>
</tr>
</tbody>
</table>

Table 5-9. Network configuration for investigating the impact of transmission range

Figure 5-14 shows true/false detection rates for different values of transmission range. As it can be seen from the figure, for shorter transmission ranges, true detection is low because the nodes detect fewer neighbours and the frequency of occurring Sybil nodes in the common list is below the threshold. For transmission range over 10 m, the true detection rate is above
98.1%. For low transmission range the chance of finding a normal node in common list is below the threshold and so the false detection rate is low. However, when the transmission range increases, the nodes detect more neighbours and hence, there are more normal nodes whose occurrence in the common list fall above the threshold and, hence, they are labelled as Sybil nodes.

Figure 5-14. True/false detection rate for different transmission ranges

5.5.6 Number of Nodes

To investigate the impact of the number of nodes on the accuracy of the proposed algorithm, I used the following configuration for the network. The number of nodes varies in the range $50 \leq N \leq 500$. The number of rounds for execution of the algorithm is 300.

Table 5-10. Network configuration for investigating the impact of the number of nodes

<table>
<thead>
<tr>
<th>Area</th>
<th>100x100 m²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes (N)</td>
<td>50-500</td>
</tr>
<tr>
<td>Malicious nodes (M)</td>
<td>5</td>
</tr>
<tr>
<td>Sybil identifiers forged by each malicious node (S)</td>
<td>10</td>
</tr>
<tr>
<td>Observer nodes (ON)</td>
<td>4</td>
</tr>
<tr>
<td>Transmission range</td>
<td>10 m</td>
</tr>
<tr>
<td>Threshold ($\theta$)</td>
<td>$0.6 \times d$</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Random Waypoint</td>
</tr>
</tbody>
</table>

Figure 5-15 shows the relation between the number of nodes and the true/false detection rates. As it can be seen from the figure, when the number of nodes increases, true detection rate also increases. This is because there are many neighbours around Sybil nodes all of which
have the ID of the Sybil node in their neighbouring list. So, the number of common nodes in the list exceeds the threshold and the Sybil node is detected.

Also, when the number of nodes increases, the false detection rate increases because a normal node may exist in the list of neighbours of so many nodes and its frequency exceeds the threshold and, as a result, it is considered as a Sybil node.

![Figure 5-15. The true/false detection rate for various numbers of nodes](image)

I further investigated the impact of the number of nodes along with other parameters on the accuracy of the algorithm. In the first step, I examined the relationship between number of nodes and number of observer nodes as per Table 5-11.

**Table 5-11. Network configuration for investigating the impact of the number of nodes and number of observer nodes**

<table>
<thead>
<tr>
<th>Area</th>
<th>100x100 m²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes (N)</td>
<td>50-500</td>
</tr>
<tr>
<td>Malicious nodes (M)</td>
<td>5</td>
</tr>
<tr>
<td>Sybil identifiers forged by each malicious node (S)</td>
<td>10</td>
</tr>
<tr>
<td>Observer nodes (ON)</td>
<td>2, 4, 6, 8, 10</td>
</tr>
<tr>
<td>Transmission range</td>
<td>10 m</td>
</tr>
<tr>
<td>Threshold (θ)</td>
<td>0.6 x d</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Random Waypoint</td>
</tr>
</tbody>
</table>
Figure 5-16 shows the impact of the number of nodes and the number of observer nodes on the true detection rate. As it can be seen from the figure, when the number of nodes increases, true detection rate also increases irrespective of the number of observer nodes. This is because the chance of finding enough critical nodes which are common neighbours of Sybil and observer nodes increases. Nodes having all these critical nodes in their neighbourhood are identified as Sybil nodes.

As per the figure, using more observer nodes is more useful when the number of nodes is small. This is because the chance of being located in the neighbourhood of Sybil nodes for observer nodes increases. When the network density is high it seems that four observer nodes are enough to achieve good performance for this network configuration.

![True Detection Rate vs. Number of Nodes](image)

Figure 5-16. The true detection rate for various numbers of nodes and number of observer nodes

Figure 5-17 shows the impact of the number of nodes and number of observer nodes on false detection rate. As it can be seen from the figure, when the number of nodes increases, false detection rate decreases irrespective of the number of observer nodes. This is because the number of critical nodes increases and so, the probability of existing a normal node in the neighbourhood list of all critical nodes decreases.
As per the figure, using more observer nodes results in better false detection rate especially when the network density is low. This is because observer nodes exchange the list of suspicious nodes with each other and so, the chance of considering a normal node as Sybil decreases.

![False Detection Rate](image)

**Figure 5-17. The false detection rate for various numbers of nodes and number of observer nodes**

In the next experiment, I studied the impact of the number of nodes and the threshold on the detection rate of the proposed algorithm. The configuration of the network is shown in Table 5-12.

**Table 5-12. Network configuration for investigating the impact of the number of nodes and the threshold**

<table>
<thead>
<tr>
<th>Area</th>
<th>100x100 m²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes (N)</td>
<td>50-500</td>
</tr>
<tr>
<td>Malicious nodes (M)</td>
<td>5</td>
</tr>
<tr>
<td>Sybil identifiers forged by each malicious node (S)</td>
<td>10</td>
</tr>
<tr>
<td>Observer nodes (ON)</td>
<td>4</td>
</tr>
<tr>
<td>Transmission range</td>
<td>10 m</td>
</tr>
<tr>
<td>Threshold (θ)</td>
<td>(0.4, 0.6, 0.8) x d</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Random Waypoint</td>
</tr>
</tbody>
</table>

![True Detection Rate](image)

**Figure 5-18 shows true detection rate for various number of nodes and different values of the threshold. As it can be seen from the figure, when the number of nodes increases, true detection rate also increases irrespective of the threshold. This is because, when the network density is high, enough critical nodes which are common neighbours of Sybil and observer nodes**
can be found with a higher probability. Nodes having all these critical nodes in their neighbourhood are identified as Sybil nodes.

As per the figure, using higher values for threshold results in higher true detection rates, especially when there are fewer nodes in the network, because there will be fewer critical nodes, and, hence, the chance of detecting the Sybil nodes which have all critical nodes in their neighbourhood list increases. However, there will also be some normal nodes having the critical nodes in their neighbourhood and so are mistakenly identified as Sybil.

![True Detection Rate](image)

**Figure 5-18. The true detection rate for various numbers of nodes and the threshold**

Figure 5-19 shows false detection rate for various number of nodes and different values of the threshold. As it can be seen from the figure, when the number of nodes increases, false detection rate decreases irrespective of the threshold. This is because the number of critical nodes increases and so, the probability of existing a normal node in the neighbourhood list of all critical nodes decreases.

As per this experiment, using higher values for threshold results in higher false detection rate especially when the number of nodes is small because the number of critical nodes decreases and the algorithm mistakenly considers more normal nodes as Sybil.
In the next experiment, I investigated the impact of the number of nodes and the number of Sybil identifiers (S) forged by a malicious node on the detection rate of the proposed algorithm. The configuration of the network is shown in Table 5-13.

Table 5-13. Network configuration for investigating the impact of the number of nodes and Sybil IDs

<table>
<thead>
<tr>
<th>Area</th>
<th>100x100 m²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes (N)</td>
<td>50-500</td>
</tr>
<tr>
<td>Malicious nodes (M)</td>
<td>5</td>
</tr>
<tr>
<td>Sybil identifiers forged by each malicious node (S)</td>
<td>5, 10, 15, 20</td>
</tr>
<tr>
<td>Observer nodes (ON)</td>
<td>4</td>
</tr>
<tr>
<td>Transmission range</td>
<td>10 m</td>
</tr>
<tr>
<td>Threshold (θ)</td>
<td>0.6 x d</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Random Waypoint</td>
</tr>
</tbody>
</table>

Figure 5-20 shows that true detection rate increases when the number of nodes increases irrespective of the number of Sybil identifiers. For lower values of S, the true detection rate is lower because spreading fewer Sybil identities causes the number of identities in the Sybil group to reach below the threshold θ and so the total Sybil group will not be detected.
Figure 5-20. The true detection rate for various numbers of nodes and Sybil IDs
As Figure 5-21 shows, increasing the number of nodes improves the false detection rate irrespective of $S$. For lower values of $S$, the false detection rate is higher because the number of critical nodes becomes smaller and hence more normal nodes are considered as Sybil.

![True Detection Rate Graph](image)

**Figure 5-21. The false detection rate for various numbers of nodes and Sybil IDs**
In the last experiment in this section, I examined the impact of the number of nodes and the number of malicious nodes ($M$) on the accuracy of the proposed algorithm. The configuration of the network is shown in

Table 5-14.
Table 5-14. Network configuration for investigating the impact of the number of nodes and malicious nodes

<table>
<thead>
<tr>
<th>Area</th>
<th>100x100 m²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes (N)</td>
<td>50-500</td>
</tr>
<tr>
<td>Malicious nodes (M)</td>
<td>2, 5, 8, 10</td>
</tr>
<tr>
<td>Sybil identifiers forged by each malicious node (S)</td>
<td>10</td>
</tr>
<tr>
<td>Observer nodes (ON)</td>
<td>4</td>
</tr>
<tr>
<td>Transmission range</td>
<td>10 m</td>
</tr>
<tr>
<td>Threshold (θ)</td>
<td>0.6 x d</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Random Waypoint</td>
</tr>
</tbody>
</table>

As Figure 5-12 shows true detection rate increases when the number of nodes increases irrespective of M. Lower values of M result in slightly better true detection rate. This is because when there are more malicious nodes in the network, it is likely that some of them located in a situation that cannot be detected by observer nodes.

![True Detection Rate](image)

**Figure 5-22. The true detection rate for various numbers of nodes and malicious nodes**

Figure 5-23 shows how false detection rate decreases when the number of nodes increases irrespective of M. Increasing M will increase false detection rate because the existence of so many malicious nodes increases the probability of a certain normal node to move with a malicious node.
Comparing Results with Similar Researches

I compared true and false detection rates of the proposed algorithm with (Jamshidi, Zangeneh, Esnaashari, & Meybodi, 2017) and (Shehni, Faez, Farshad, & Kelarestaghi, 2017). These two works have been chosen for comparison of the results because of their similarity in the attack model and the approach with my work. In this experiment, the same network configuration has been used for all three methods, which is detailed in Table 5-15. The number of rounds for execution of the algorithm is 300. I repeated the experiments 100 times and used the average of the results.

<table>
<thead>
<tr>
<th>Area</th>
<th>100x100 m²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes (N)</td>
<td>300</td>
</tr>
<tr>
<td>Malicious nodes (M)</td>
<td>5</td>
</tr>
<tr>
<td>Sybil identifiers forged by each malicious node (S)</td>
<td>10</td>
</tr>
<tr>
<td>Observer nodes (ON)</td>
<td>4</td>
</tr>
<tr>
<td>Transmission range</td>
<td>10 m</td>
</tr>
<tr>
<td>Threshold (θ)</td>
<td>0.6 x d</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Random Waypoint</td>
</tr>
</tbody>
</table>

Table 5-15. Network configuration for comparing the detection rated of the proposed algorithm with (Jamshidi et al., 2017) and (Shehni et al., 2017)

Figure 5-24 compares the average true detection rate of the proposed algorithm with the other two algorithms.
Figure 5-24. Comparing the true detection rate of the proposed algorithm with similar researches

Figure 5-25 compares the average false detection rate of the proposed algorithm with the other two algorithms.

Figure 5-25. Comparing the false detection rate of the proposed algorithm with similar researches
As it can be seen from the above two diagrams the proposed algorithm has a higher average true detection rate and lower false detection rate than the other two algorithms. This is because in the proposed algorithm observer nodes communicate with other nodes to know about their neighbours. In this way, the algorithm gets a more holistic picture of what is happening in the network. Note that high accuracy is achieved in the proposed algorithm without sacrificing the efficiency (as it was already discussed the proposed algorithm has a lower overhead than the other two algorithms).

Although the algorithms discussed in Chapter 3 for static WSN cannot be formally compared with the proposed algorithm, reviewing their results (see Figure 3-12 and Figure 3-13) with my results reveals this fact that none of those algorithms had better true and false detection rates than my algorithm at the same time.

For the last experiment, I compared the scalability of the proposed algorithm with (Jamshidi, Zangeneh, Esnaashari, & Meybodi, 2017) and (Shehni, Faez, Farshad, & Kelarestaghi, 2017). To do so, I used the following configuration for the network. The number of nodes varies in the range $100 \leq N \leq 1000$. The number of rounds needed for each algorithm to converge is measured, i.e., the number of rounds after which the algorithm does not show a significant improvement in true/false detection rate.

<table>
<thead>
<tr>
<th>Area</th>
<th>100x100 m²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes (N)</td>
<td>100 - 1000</td>
</tr>
<tr>
<td>Malicious nodes (M)</td>
<td>5</td>
</tr>
<tr>
<td>Sybil identifiers forged by each malicious node (S)</td>
<td>10</td>
</tr>
<tr>
<td>Observer nodes (ON)</td>
<td>4</td>
</tr>
<tr>
<td>Transmission range</td>
<td>10 m</td>
</tr>
<tr>
<td>Threshold (θ)</td>
<td>0.6 x d</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Random Waypoint</td>
</tr>
</tbody>
</table>

Table 5-16. Network configuration for comparing the scalability of the proposed algorithm with (Jamshidi et al., 2017) and (Shehni et al., 2017)

Figure 5-26 shows the relation between the number of nodes and the number of rounds needed for each algorithm to converge. As it can be seen from the figure, when there are more nodes in the network, the other two algorithms need more rounds for convergence. This is while, the number of rounds needed by the proposed algorithm to converge does not increase significantly. Therefore, the proposed algorithm shows better scalability than (Jamshidi et al., 2017) and (Shehni et al., 2017).
Chapter Summary

In this chapter, I evaluated the proposed algorithm in terms of overhead (memory, communication and computation) and true/false detection rates. The analysis of the experimental results revealed that the proposed algorithm does not impose a significant overhead. Comparing with similar researches, the proposed algorithm has a lower overhead.

I investigated the impact of several factors on the detection rate of the proposed algorithm. These factors included the number of nodes, the number of malicious nodes, the number of Sybil identities forged by each malicious nodes, the number of observer nodes, the threshold and the transmission range. The results are summarised as follows:

- **Number of nodes:** When the number of nodes increases, both true and false detection rates increase.
- **Malicious nodes (M):** M does not have a significant impact on true detection rate. However, increasing M increases false detection rate.
- **Sybil identifiers (S):** A lower value of S results in a lower true detection rate. However, the impact of S on false detection rate is negligible.
- **Observer nodes (ON):** Using more observer nodes increases true detection rate and decreases false detection rate. However, using more observer nodes imposes more overhead. The experimentation showed that for the designated network, using only four observer nodes are enough for the algorithm to achieve good results.
• Threshold (θ): Lower values for θ results in higher true and false detection rates.
• Transmission range: For shorter transmission ranges, both true and false detection rates are low.

Finally, I compared the detection rate of the proposed algorithm with (Jamshidi, Zangeneh, Esnaashari, & Meybodi, 2017) and (Shehni, Faez, Farshad, & Kelarestaghi, 2017). The results showed that the proposed algorithm outperforms the other two algorithms in terms of overhead, and true/false detection rates.
Chapter Six: Conclusions and Future Work

This chapter concludes the thesis and highlights the main contributions and findings. The first section summarises the thesis and the final section discusses the limitations of the study and provides a list of opportunities for further research.

6.1 Summary

Wireless sensor networks (WSN) consist of a number of sensor nodes which are spread in an environment to collect data for a specific application (e.g., military, medical, meteorological, environmental, and industrial process monitoring). Sensor nodes communicate with each other and with the base station through a wireless network. The main power source of a sensor node is its battery which is not chargeable and when finishes the node stops all its functionalities. In some WSNs nodes are mobile and may change their locations over time. This type of WSN is called mobile wireless sensor network (MWSN).

To reduce the cost of employing a WSN for an application, sensor nodes are fabricated with limited capabilities: battery, memory, processor, and communication bandwidth. Because of these limitations, such networks are susceptible to many types of attacks. There are several known attacks against WSN: wormhole, black hole, sinkhole, blackmail, and Sybil, to name a few. Since Sybil attack is one of the popular and devastating attacks against WSN, this research was mainly focused on detecting this attack in MWSN.

There have been many researchers conducted on detecting this attack in static WSN, which mostly rely on the location of nodes or their signal strengths and hence are unsuitable for MWSN. There are also a number of researches conducted on detection of Sybil attack in MWSN some of which were reviewed in this thesis. Some of these algorithms suffer from a significant overhead and some of them suffer from low accuracy.

In this thesis, a new algorithm was proposed for detecting the Sybil attack in MWSN. The algorithm uses observer nodes, which are normal trustful nodes and are programmed especially to collect some information about their neighbours and exchange this information with each other to identify suspicious nodes. The algorithm relies on this fact that Sybil nodes have the same neighbouring nodes. Thus, nodes appear in many neighbouring lists are labelled as critical and nodes having all these critical nodes in their neighbouring lists are considered as suspicious. The observer nodes exchange the list of their suspicious nodes to identify Sybil nodes.
The analysis of the algorithm showed its low complexity in terms of memory, communication, and computation overhead. The simulation results showed the algorithm achieved better detection rates than similar researches. In addition, the impact of several factors on the detection rate of the proposed algorithm has been investigated which include: number of Sybil identities forges by each malicious nodes, number of malicious nodes, number of observer nodes, threshold, number of nodes, and transmission range. The following facts inferred from the results:

- The number of Sybil identities forged by a malicious node does not have a significant impact on the false detection rate. However, creating more Sybil identities increases true detection rate. When malicious nodes spread fewer Sybil identities, the occurrence of critical nodes in the common neighbours of the observer nodes and their neighbours falls below the threshold and hence the Sybil nodes cannot be detected accurately.
- The impact of the number of malicious nodes on the true detection rate is negligible. However, having more malicious nodes results in higher false detection rate in the initial rounds of the algorithm. This is because the existence of so many malicious nodes increases the probability of a certain normal node to move with a malicious node during the initial several rounds of the algorithm. This problem gets fixed after the algorithm progresses.
- Using more observer nodes increases true detection rate and decreases false detection rate. However, using more observer nodes imposes more overhead. The experimentation showed that for the designated network, using only four observer nodes are enough for the algorithm to achieve good results.
- Using lower values for the threshold results in higher true and false detection rates. If the threshold is considered too high, in the case that malicious nodes generate just few Sybil IDs, or their neighbouring nodes move quickly, the frequency of their neighbours in the common lists falls below the threshold and cannot be considered as critical. Therefore, the Sybil nodes cannot be accurately detected by the algorithm.
- When the network is condensed (i.e., there are many nodes in the network) both true and false detection rates are high. This is because a normal node may have so many neighbours and its frequency exceeds the threshold and, hence, it is mistakenly considered as a Sybil node.
- For shorter transmission ranges, both true and false detection rates are low. This is because the nodes detect fewer neighbours and hence, the frequency of occurring the
neighbours of Sybil nodes in the common lists falls below the threshold and the Sybil nodes cannot be accurately detected.

The memory overhead of the proposed algorithm is \(O(ON \times n)\) where \(ON\) is the number of observer nodes and \(n\) is the total number of nodes in the network. Its complexity overhead is \(O(ON \times p)\) where \(p\) is the number of rounds the algorithm executes. The computation of the proposed algorithm is \(O(n \times ON \times p)\). Comparing with Jamshidi et al., (2017) and Shehni et al., (2017), the proposed algorithm imposes lower overhead.

The average true detection rate of the proposed algorithm is 98.1% while for Jamshidi et al., (2017) is 91% and for Shehni et al., (2017) is 95.4%, respectively. The average false detection rate of the proposed algorithm is 0.5% while for Jamshidi et al., (2017) is 1.2% and for Shehni et al., (2017) is 4.3%, respectively. Therefore, the proposed algorithm outperforms these algorithms not only in terms of overhead but also in terms of accuracy.

Number of rounds needed for the proposed algorithm to converge has a linear relationship with the number of nodes with a small slope. In Jamshidi et al., (2017) and Shehni et al., (2017) the correlation between number of rounds and number of rounds is also a linear function but with a larger slope than the one in the proposed algorithm. Therefore, the proposed algorithm is more scalable than the other two algorithms.

6.2 Future Work

In the proposed algorithm, the threshold is fixed throughout the network lifetime. However, according to the results of experimentations, to achieve high accuracy, it must be changed appropriately based on other factors such as transmission range, number of nodes, number of Sybil IDs, number of observer nodes, and number of malicious nodes. One difficulty in this regard is collecting the information about the aforementioned factors dynamically, especially when the nodes are mobile. There are decentralised algorithms for measuring network parameters, which can be used for this purpose. Incorporating machine learning algorithms in the proposed algorithm for tuning the threshold parameter efficiently can significantly improve the accuracy of the algorithm.

In the experimentations, the optimum value of the observer nodes for the designated network was four. There is some space for theoretical analysis on this and formulating the optimum number of observer nodes in terms of other factors of the network.
One risk for the proposed algorithm is its assumption on the trustworthiness of the observer nodes. If they become captured by adversaries, the algorithm may fail. Thus, one avenue for the future research could be lifting up the security level of the observer nodes themselves.

In the simulation, Two Ray Ground has been considered as the physical model and AODV as the protocol. One avenue for the future research would be investigating the impact of the physical model and the protocol on the performance of the proposed approach.
References


Declaration

Name of candidate: Abdulaziz Abdulrahman Alomirah

Principal Supervisor: Bahman Sarrafpour
Associate Supervisor/s: Xiaosong Li

CANDIDATE’S DECLARATION

I confirm that:

- This Thesis/Dissertation/Research Project represents my own work;
- The contribution of supervisors and others to this work was consistent with the Unitec Regulations and Policies.
- Research for this work has been conducted in accordance with the Unitec Research Ethics Committee Policy and Procedures, and has fulfilled any requirements set for this project by the Unitec Research Ethics Committee.

Research Ethics Committee Approval Number:

Candidate Signature: ____________________________ Date: 27/09/2018

Student number: 1396908
Full name of author: Abdulaziz Abdulrahman Alomirah

ORCID number (Optional):

Full title of thesis/dissertation/research project ('the work'):
Detecting Sybil Attack in Mobile Wireless Sensor Networks Using Observer Nodes

Practice Pathway: .................................................................................................................................

Degree: Master of Computing
Year of presentation: 2018

Principal Supervisor: Bahman Sarrafpour
Associate Supervisor: Xiaosong Li

Permission to make open access
I agree to a digital copy of my final thesis/work being uploaded to the Unitec institutional repository and being made viewable worldwide.

Copyright Rights:
Unless otherwise stated this work is protected by copyright with all rights reserved.
I provide this copy in the expectation that due acknowledgement of its use is made.

AND

Copyright Compliance:
I confirm that I either used no substantial portions of third party copyright material, including charts, diagrams, graphs, photographs or maps in my thesis/work or I have obtained permission for such material to be made accessible worldwide via the Internet.

Signature of author: ..............................................
Date: 27/09/2018