An enhanced security model for wireless sensor networks.

An enhanced security model for Wireless Sensor Networks

by

Puseletso Sebothoma

(205124837)

Dissertation submitted in partial fulfilment of the requirements for the degree

MComp: Information Technology

in the

Department of Information Technology

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

at the

TSHWANE UNIVERSITY OF TECHNOLOGY

Supervisor: Dr. T.E Mathonsi

Co-Supervisor: Dr. D.P du Plessis

June 2023

DECLARATION

I, Puseletso Sebothoma, hereby confirm that this research study entitled: An enhanced security model for Wireless Sensor Networks, submitted for the degree Master of Computing (MComp): Information Technology at the Tshwane University of Technology is my research and has not been submitted to any other institution of higher learning. I further confirm that all articles quoted are indicated and acknowledged using a comprehensive list of references.

Signature: Date: May 2023

DEDICATION

The study is dedicated to my wife, kids, and family members who guided, supported, and encouraged me to complete my studies.

ACKNOWLEDGMENTS

First and foremost, I would like to thank the Almighty God, who gave me the power and guidance to finish this research study. I also wish to express my heartfelt gratitude for the kindness of my supervisors, who contributed much to my completion of this research study (Supervisors: Dr. T.E Mathonsi and Dr. D.P du Plessis). These mentors continued to give me professional advice such as focusing on the research topic, and they provided me with good feedback on my research study.

Lastly, I would like to give thanks to family, friends, and colleagues, for the willing support and motivation that they showed me during the period of this research study.

PUBLICATIONS

Parts of this thesis have been published and presented in the following journals and conference proceedings:

Journals:

1. Puseletso Sebothoma and Topside E. Mathonsi, “Implementation of an Enhanced Security Algorithm for Wireless Sensor Networks”. Submitted to I.J. Computer Network and Information Security, October 2023.

Conference proceedings:

1. Puseletso Sebothoma and Topside E. Mathonsi, “An Enhanced Security Algorithm for Wireless Sensor Networks”. Accepted for publication at the 3rd International Conference on Electrical, Computer and Energy Technologies, Cape Town, South Africa, 16-17 November 2023.

ABSTRACT

In recent years, Wireless Sensor Networks (WSNs) have been widely used to collect and transmit data. Because these networks are inexpensive and allow easy communication, they are used in a variety of surveillance applications today. The high bandwidth requirements, high power consumption, quality of service (QoS) delivery, data processing and compression techniques, makes the WNSs to be prone to Denial-of-Sleep (DoSL) attacks, and makes it difficult for WSNs to function well. During the DoSL attacks unnecessary fake packets are sent, which completely drains the power supply, thereby shortening the lifespan of the nodes. In addition, these attacks reduces the overall performance of the network. In order to address the identified gaps, this research study integrated existing models, namely, S-MAC protocol, AES algorithm and RSA algorithm when designing the proposed DSD-RSA algorithm. The DSD-RSA algorithm was developed to minimize power consumption and end-to-end delay while maximizing network throughput in WSNs. The Network Simulator-2 (NS-2) was used to simulate and evaluate DSD-RSA against WSN-FAHN and GA-DoSLD. The proposed DSD-RSA algorithm, GA-DoSLD, WSN-FAHN, and GSHMAC algorithms experienced an average throughput of 95%, 89%, and 60%, an average end-to-end delay of 19%, 35%, 40%, and 60%, and average energy consumption of 30%, 35%, 45%, and 50%, respectively. Thus, the proposed DSD-RSA algorithm will find applications in industries, society, and academia due to its superior performance.

Keywords —DSD-RSA, Denial of Sleep attacks, RSA Cryptography, WSN

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF ACRONYMS

AES: Advanced Encryption Standard CSMA: Carrier Sense Multiple Access

DARPA: Defense Advanced Research Projects Agency

DoSL: Denial of Sleep

DSD: Dysfunction Sensor Detection

GPS: Global Positioning System

GSHMAC: Green and Secure Hybrid Medium Access Control

MANET: Mobile ad hoc Network

MATLAB: Matrix Laboratory

N-2: Network Simulator-2

NS-3: Network Simulator-3

OMNET: Optical Micro-Networks

RSA: Rivest Shamir Adleman

SensIT: Sensor Information Technology

T S-MAC: Sensor MAC

TDMA: Time Division Multiple Access

WSNs: Wireless Sensor Networks

UWBs: Ultrawide Bands

Chapter 1: Introduction

1.1 Background Information to the Study

Wireless Sensor Networks (WSNs) are composed of several spatially distributed nodes, connected to one or more sensors, which monitor a large physical environment (Farraposo, 2014). The nodes (wireless devices) are typically small in size and capable of performing sensing, onboard processing, communication, and storage. WSNs offer economically viable solutions for a variety of applications such as current implementation to monitor factory instrumentation, pollution levels, freeway traffic, and the structural integrity of buildings.

The development of WSNs has resulted mainly from military applications such as battlefield surveillance. In 1978, the Defense Advanced Research Projects Agency (DARPA) organized the Distributed Sensor Nets Workshop, focusing on sensor network research challenges such as networking technologies, signal processing techniques, and distributed algorithms. DARPA also operated the Distributed Sensor Networks (DSN) programme in the early 1980s, which was followed by the Sensor Information Technology (SensIT) programme. At the moment WSNs are viewed as one of the most important technologies of the 21st century (Zheng et al., 2015).

There are still issues, such as high bandwidth demand, high energy consumption, that have to be solved before WSNs are widely deployed. Because of their security vulnerabilities, WSNs are unprotected from attacks by malicious nodes (Azzabi et al., 2017). Therefore, the research must implement an enhanced security model that can detect intruders in WSNs. The following are some of the causes of insecure WWSNs:

a) Wireless Medium

The wireless medium is inherently less secure because its broadcast nature makes eavesdropping simple. Any transmission can easily be intercepted, altered, or replayed by an adversary. The wireless medium also allows the attacker easily to intercept valid packets and easily to inject malicious packets (Farraposo, 2014).

b) Hostile Environment

The hostile environment is the next challenging factor in which sensor nodes function. It is fair to say that nodes face the possibility of destruction or capture by attackers. The highly hostile environment represents a serious challenge for security researchers (Zheng et al., 2015).

WSNs have become more important as they can be found in research projects and civilian applications, as well as defence projects. Benefits go to the service provider resulting from low cost in network deployment and end users with ubiquitous access to the internet from anywhere at any time (Mukherjee & Mukherjee, 2019). The sensor nodes are often deployed in remote and inaccessible areas, thereby increasing their exposure to malicious intrusions and attacks. Many issues have therefore to be addressed before WSNs are widely deployed and trustworthy. WSNs face several security challenges when deployed in remote areas. One of the most challenging security threats is Denial of Sleep (DoSL). DoSL attacks in wireless sensor networks (WSNs) refer either to intentional or accidental actions that prevent sensor nodes from entering into a low-power sleep mode, leading to the depletion of their energy reserves. In WSNs, energy efficiency is a critical factor, as the sensors are usually battery-powered and deployed in remote or hard-to-reach locations (Sicari et al., 2018; Gavric & Simic 2018; Premkumar & Sundararajan, 2020). A DoSL attack can be defined as a cyber-attack in which the perpetrator seeks to make a machine or network

resource unavailable to its intended users by temporarily or indefinitely disrupting the services of a host connected to the internet. DoSL affects WSNs because it makes it difficult for the users to access the network, and to have their private information secured. DoSL is typically accomplished by flooding the targeted machine or resource with superfluous requests in an attempt to overload systems and prevent some or all legitimate requests from being fulfilled (Desai et al., 2017). DoSL can compromise the availability of WSNs because such attacks could essentially prevent one or more nodes from accessing or providing specific services.

1.2 Motivation of the Study

The literature review has shown that in order to enable users to have a better and higher-quality service, the security model should be designed in a way that it should improves the network performance of WSNs, and also by reducing DoSL attacks. DoSL attacks make it difficult for the network to function as expected; this is why creating a model to prevent attacks is essential. I therefore took it upon myself to find a solution that could minimise these attacks, making the WSNs user-friendly. Furthermore, continuous learning has been the driving factor in my career and studies. In doing this research, I wanted to gain more and deeper knowledge of the WSNs applications, and of the multiple DoSL attacks that exist in WSNs. I have an interest in creating and developing my research skills and in gaining more experience of how to solve problems professionally and logically; such thinking further motivated this investigation.

1.3 Problem Statement

A DoSL attack takes advantage of high bandwidth demand, high energy consumption, quality of service (QoS) provisioning, data processing and compressing techniques, and cross-layer design, making it difficult for WSNs to function well. WSNs need

power limitation of the sensor nodes; they are scheduled occasionally to go into sleep mode − DoSL attacks prevent the nodes from transiting to sleep mode. This is done by sending unnecessary fake packets and by leaving their radio turned on (trying to process the fake packets) thereby completely exhausting the power supply. This reduces their lifetime from years to months, or months to weeks, or even weeks to days. By these means, the overall network performance is reduced. Therefore, WSNs call for more effective and applicable solutions (Hussain, 2020). WSNs offer economical, flexible, scalable, and pragmatic solutions to many situations. Sensor nodes are tiny and have a limited, non-rechargeable battery source, small memory abilities, and low transmitter power.

1.4 Research Questions

The main research question that this study addresses reads as follows: How does one develop a mechanism that secures the WSNs from DoSL attacks?

There are sub-questions that must be answered to provide a better understanding of WSNs as a whole and how they work. The following three sub-questions addressed in this research are:

1. What are the challenges and limitations of the existing security models currently used to protect and secure WSNs from DoSL attacks?
2. How can an enhanced security model be designed to detect DoSL attacks in WSNs?
3. What is the effectiveness of the proposed enhanced security model versus the state-of-the-art-related algorithms in WSNs?

1.5 Aim and Objectives of the Study

The main aim of this research study was to develop a mechanism that secures WSNs against DoSL attacks. The mechanism developed was to improve two-factor user authentication which is so far one of the most reliable authentications in WSNs, and helps significantly in preventing DoSL attacks from affecting the functioning and good performance of WSNs. Below are the technical objectives we used.

1. To identify and analyse the challenges and limitations of the existing security schemes currently being used to secure and protect WSNs from DoSL attacks.
2. To design an enhanced security model that can detect DoSL attacks in WSNs.
3. To evaluate the effectiveness of the proposed enhanced security model versus the state-of-the-art related algorithms in WSNs.

1.6 Research Methodology

Research methodology a scientific processes, procedures or techniques that is used in the identification, selection, processing, synthesis, and analysis of information about a specific phenomenon under investigation. The research methodology deals furthermore justifies the research reliability and validity. There are two main type of main research methodologies used in scientific research, namely experimental research methodology and design science research methodology. The purpose of experimental research methodology is to collect data that can be used to accept or refuse to accept a hypothesis. Experiments are normally used because they are cost effective. They most times involved coming up with a simulator and evaluate parameters in a lab environment or in a simulated environment before coming up with a product. In addition, experimental research designs are typically well-defined and standardized, making them highly replicable. This means that other researchers can

repeat the same experiment under similar conditions, increasing the confidence in the findings. Additionally, well-designed experiments often produce generalizable results that can be applied to a wider context. Design science research methodology involves the creation/development of an artifact and/or design theory to improve the current state of practice as well as existing research knowledge. In this research methodology a prototype/artifact must be produced. As a result, this study adopted an experimental research methodology in order to achieve the main aim of the study with the research methods presented in Figure 1.1.

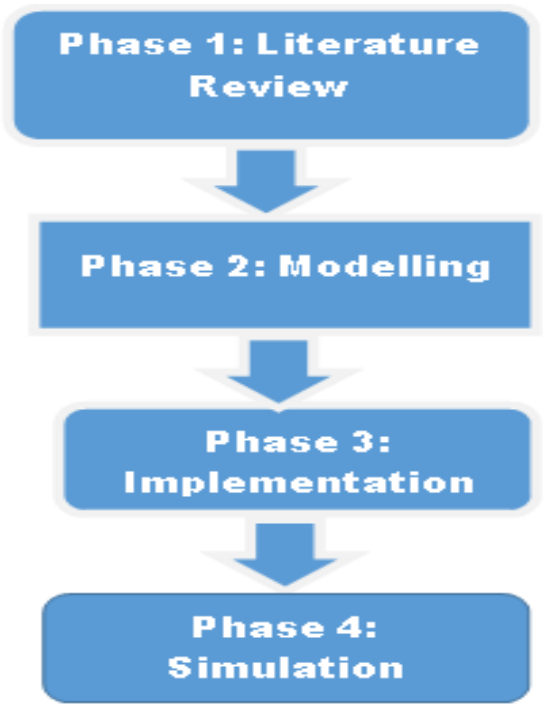


Figure 1.1: Research methods phases

1.6.1 Research methods phases

1.6.1.1 Phase 1: Literature review

Information searched for on internet sources, related workbooks, published journals, conference papers, and from other sources was used for the purpose of gaining more knowledge on this study.

1.6.1.2 Phase 2: Modelling

The scientific and mathematical models were used to develop the mechanism proposed to secure WSNs.

1.6.1.3 Phase 3 and 4: Implementation and simulation

Network Simulator (NS-2) has been used for the development and implementation of the proposed DSD-RSAalgorithm. NS-2allows researchers to experiment and implement newly designed models in a controlled manner.

1.7 Contribution of the Dissertation

The dissertation includes the following:

1. Analysing and evaluating the performance of the developed mechanism under DoSL attacks can ensure that the data received is from a known and trusted source. This will improve the network performance and minimise the DoSL attacks from destroying the WSNs performance.
2. The development of a security model has ensured the speed and efficiency of the proposed algorithm when authenticating WSNs users. This was achieved through the use of the two-factor user authentication

algorithm which is so far one of the most reliable and most used authentications to prevent DoSL attacks from affecting WSNs.

3. The proposed security model prevented malicious users from connecting with users on the network. This prevents DoSL attacks from attacking users to use WSNs freely.

1.8 The Dissertation Structure

The remainder of the study will be structured as in Figure 1.2.

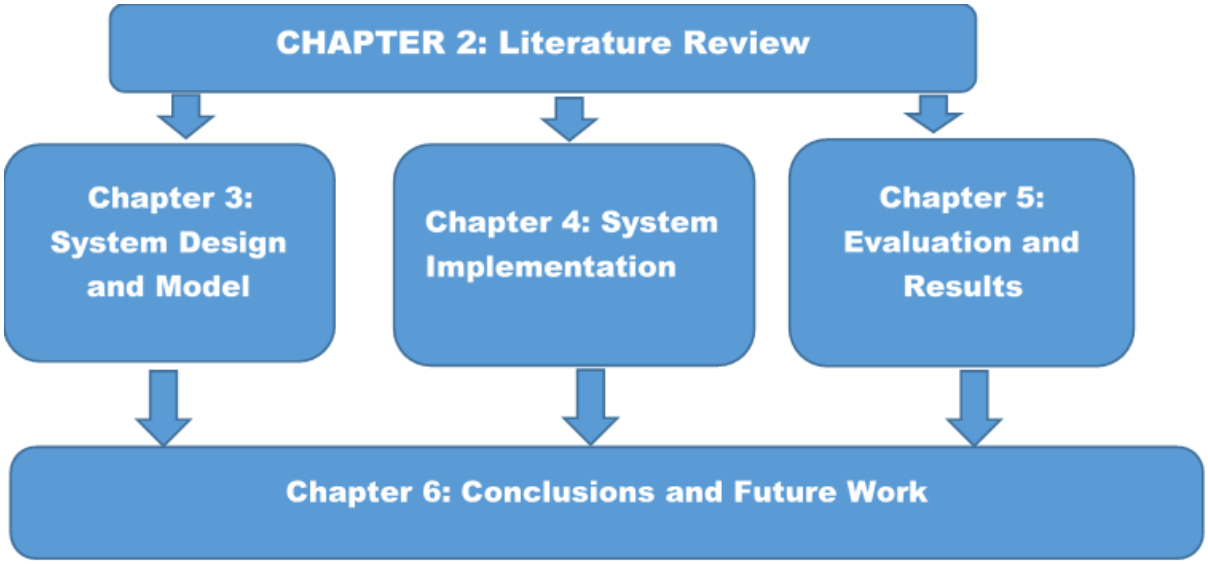


Figure 1.2: Dissertation structure

The following chapter, which is Chapter 2, presents how the literature review was conducted. This chapter presents an overview of WSNs, DoSL attacks, and all types of attacks in WSNs. Furthermore, the chapter explores more about the related work that has been provided to give a clear indication of what has been achieved in the field of WSNs and DoSL attacks. Chapter 3 presents the proposed algorithm. Chapter 4 presents various network simulators. Chapter 5 discusses and presents the performance and results of the proposed mechanism developed for WSNs. Chapter 6 presents the conclusion and offers future work for the research study.

1.9 Chapter Summary

In this chapter, an overview of the WSNs was presented. The chapter further presented the problem statement, research questions, aim and objectives. The motivation and aim of this dissertation was to develop a mechanism that would detect DoSL attacks from WSNs. The journals, books, internet, among many other sources were identified as sources in answering the research questions. The literature review and related work are discussed in the next chapter.

Chapter 2: Literature Review

2.1 Introduction

WSN has been one of the dominant networks for a very long time. According to stats, 49% of organizations state that keeping up with security is now more difficult than ever, especially in the multi-cloud world (Ayaz et al., 2017). As with the network such as cloud computing, mobile ad hoc network (MANET), among others, not being well secured, so WSNs are equally vulnerable. The challenges found in WSNs are the security, routing protocols, power administration, scalability, among others (Chan et al., 2020). One of the major attacks in WSNs is the DoSL attack, on which our research focuses (Fotohi et al., 2020). DoSL affects WSNs because its main aim is the disruption of service by attempting to limit access to a machine or service instead of subverting the service itself (Patil & Chaudhari, 2016). DoSL achieves its goal by sending a victim a stream of packets that swamps his network or processing capacity, denying access to its regular clients. WSNs make no use of cables for connections it uses radio frequency. One of the greatest advantages of WSNs is their mobility, and easy connection (Fotohi et al., 2020). It is therefore important to develop a mechanism that improves the security of WSNs. Due to security issues already mentioned and limited resources and energy, WSNs are vulnerable to attacks.

Sensor nodes must not disclose their data to the neighbouring nodes. However, attackers can access, and ultimately gain information that should be confidential to the users. Improving the connectivity of WSNs or of an Internet of Things (IoT) network is one of the main ways of improving quality of service (QoS) in terms of communication, target detection, and cyber-security (Singh et al., 2020). In general, this kind of problem arises as a gap or a void in the case of a random deployment of the network.

All the existing methods present solutions avoiding gaps or voids; however, detection of these voids is not always shown.

In this chapter, we will discuss and illustrate issues or matters related to WSNs. The chapter will further discuss wireless security, our research study being closely related to such. Furthermore, we present the advantages and disadvantages of WSNs. In addition, various types of encryption are discussed.

2.2. Overview of Wireless Networks

The wireless network can either be an indoor or an outdoor network. Unlike a wired network that uses devices such as laptop to desktop computers, a wireless network allows devices to remain connected to the network; however, they may roam untethered to any wired device (Sadiku & Akujuobi, 2022). Access point or wireless fidelity (Wi-Fi) works by connecting directly to your broadband router or network switch with a data cable or Ethernet (Gabriel & Onuodu, 2021). It will then transmit and receive a wireless signal.

With WSNs, network resources can be accessed from any location within its coverage area or from any Wi-Fi hotspot. Security advances in a wireless network provide robust security protocols. WSNs can also cost less because they eliminate or reduce wiring expenses. They can cost less to operate than wired networks (Bi et al., 2016). A WSN is easy to set up; no cables must be strung, making installation quick and cost-effective (Khan et al., 2016). A wireless network allows devices to be linked to the network yet to roam without any cables. WSNs are advantageous to use because users can move anywhere without being restricted to a single location (Juma, 2020).

There are two types of wireless networks: infrastructure-based, and infrastructure-less (Rajesh et al., 2022). In infrastructure-based networks, networking devices such as

access points and base stations are located throughout the network. Mobile nodes can establish a network connection by disconnecting from the base station range and connecting to another base station (Fatima & Khursheed, 2022). However, in infrastructure-less networks, devices are connected without fixed base stations and access points. Dynamically, the mobile node is built and maintained to operate as a router. Figure 2.1 shows the image of the wireless router network.

Figure 2.1: Wireless router network (Fatima & Khursheed, 2022)

An indoor positioning system is a network of devices used to locate people or objects where GPS and other satellite technologies lack precision or fail entirely; such as inside multi-story buildings, airports, parking garages, or even underground locations (Nguyen et al., 2020). This network is susceptible to environmental distortion attenuation of the signal, which can affect positioning accuracy (Mahmud, 2022).

Indoor location-based services (LBSs) are expanding rapidly. A typical indoor wireless network is depicted in Figure 2.2.

Figure 2.2: Indoor wireless network (Mahmud, 2022)

Outdoor Wi-Fi is simply when Wi-Fi access is provided outside of a building. It is usually used as an extension of the internal wireless LAN network of an organization, to give users a continuous Wi-Fi service (Tian et al., 2021).

The GPS is widely used in outdoor WSNs. We can say clearly that there is not much difference between indoors and outdoors. There is also not much operational difference between these WSNs (Yan, 2015). Figure 2.3 present the overview of the outdoor wireless network:

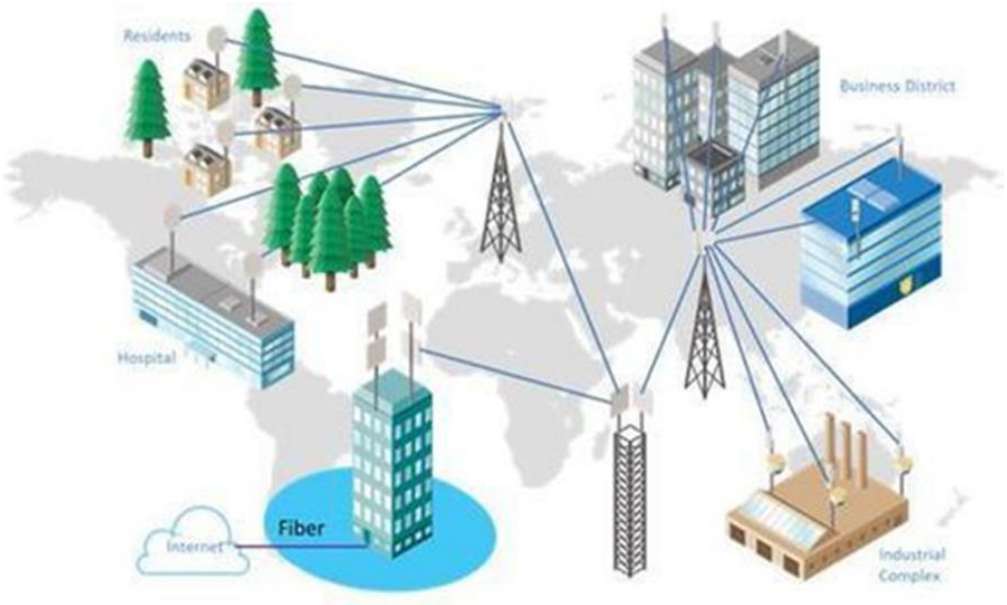


Figure 2.3: Outdoor wireless networks (Tian et al., 2021)

2.3. Overview of Wireless Sensor Networks

WSNs can be defined as self-configured and infrastructure-less wireless networks which monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants. WSNs cooperatively pass their data through the network to the main location, as depicted in Figure 2.4 (Maatin, 2012). WSNs can also be referred to as networks of spatially dispersed sensors dedicated to monitoring and recording the physical conditions of the environment, forwarding the collected data to a central location. This chapter gives the overview of WSNs; it should always be borne in mind that the enhanced model must be created to secure WSNs. Hast (2015) describes WSNs as a group of specialized transducers with a communications infrastructure for monitoring and recording conditions at the diverse locations. A WSN is a self-configuring wireless network comprising numerous sensors deployed in an ad hoc manner to monitor system conditions and physical or environmental parameters. These sensors are connected to base stations (BSs), which serve as processing units within the WSN. The medium access control (MAC)

system is one of the major concerns when designing WSNs. One of the primary causes of excessive energy consumption or wastage is the transfer of data by the two nodes sharing the same medium at the same time (Yen, 2012). This data leads to the collision of the data packets.

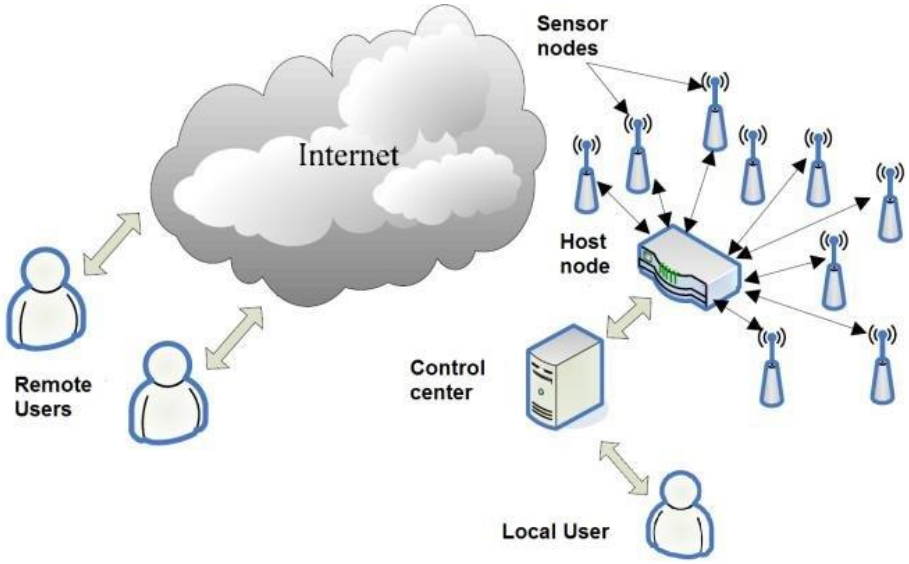


Figure 2.4: Typical wireless sensor network (Maatin, 2012)

2.3.1 Classification of WSNs

The WSNs uses wireless data connections between network nodes. WSNs describe telecommunication networks whose interconnections have been accomplished by enabling data exchange between geographically dispersed sensors (Ding et al., 2014). A WSNs can also be defined as a self-configured and infrastructure-less wireless network for monitoring physical or environmental conditions such as pressure, motion, and pollutants. The research will also be considering ZigBee, currently in smart grids, the most widely used WSNs protocol. ZigBee is also known as (IEEE 802.15.4), which deals with the upper network and application layers. ZigBee is designed for local networks in home environments: it cannot directly communicate to servers on the internet. ZigBee will help in lowering power and adding to long battery life

communication technologies, allowing low data-rate devices to communicate with one another. Ultrawide bands (UWBs) can also help in lowering battery life and saving power. Such could reduce the time it takes to transfer authenticated patient data from the healthcare server to the user, with standards such as Wi-Fi and Bluetooth. Wireless sensor network protocols are depicted in Figure 2.5.

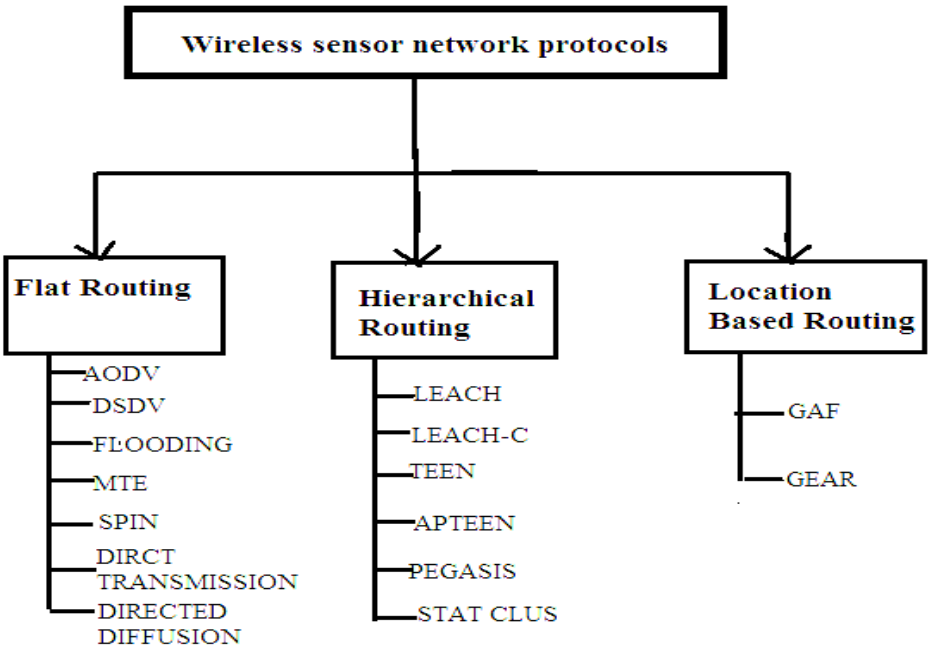


Figure 2.5: Overview of WSNs (Ding et al., 2014)

WSNs can be classified into several categories based on various criteria, including the following:

1. Deployment:
    a. Homogeneous WSNs: These are networks in which all the nodes have the same capabilities and characteristics (Rostami et al., 2018). This type of WSNs is commonly used in applications in which the environment and requirements are uniform throughout the network.
    b. Static deployment: involves manually placing sensor nodes in specific locations. This method is typically used for applications where the sensor

network's topology is fixed and does not need to change over time. For example, static deployment is often used in industrial settings to monitor equipment or in environmental monitoring to track pollution levels (Rostami et al., 2018; Vyas et al., 2020).

c. Random deployment: involves randomly placing sensor nodes within a predefined area. This method is often used for applications where the sensor network's topology needs to be flexible and adaptable. For example, random deployment is often used in disaster relief to monitor damaged infrastructure or in wildlife tracking to monitor animal movements (Rostami et al., 2018; Vyas et al., 2020).

2. Application:

a. Environmental Monitoring WSNs: These are networks used for monitoring various environmental parameters such as temperature, humidity, and atmospheric pressure (Vyas et al., 2020).

b. WSNs for industrial process monitoring are networks utilized to observe industrial processes, including temperature, pressure, and flow rate (Priyanka et al., 2021). On the other hand, the health monitoring WSNs is a network that monitors the wellness of individuals and provides real-time feedback.

3. Network Structure:

a. Flat WSNs are networks in which all nodes are on the same hierarchical level, and there is no predefined structure for the network topology (Chan et al., 2020). Flat WSNs are commonly used in applications where the sensing environment is simple, and network nodes are expected to collaborate equally in data processing and communication tasks (Abdalkafor & Aliesawi, 2022).

b. Hierarchical WSNs: These are networks in which nodes are organized into various levels of hierarchy, with some nodes serving as gateways or cluster heads (Mehta & Saxena, 2020).

4. Communication Protocol:

   a. Single-Hop WSNs: These are networks in which communication is limited to a single hop or direct communication between nodes (Hossain et al., 2020).

   b. Multi-Hop WSNs: These networks facilitate communication between source and destination nodes through intermediate nodes, also referred to as hops, thereby enabling the exchange of information over extended distances (Sakib et al., 2022).

5. Energy Consumption:

   A. Energy-efficient WSNs: These are networks designed to minimize energy consumption and prolong the lifespan of the nodes (Nivedhitha et al., 2020).

   B. Energy-constrained WSNs: These are networks in which the energy supply of the nodes is limited and must be conserved (Chowdhury & Hossain, 2020).

2.3.2 Benefits and challenges of having Wireless Sensor Networks

2.3.2.1 Benefits of WSNs:

WSNs offer several benefits, including:

1. Low cost: WSNs are relatively inexpensive to deploy and maintain compared with wired sensor networks (Appiah et al., 2020). This is because the sensors are small and battery-powered, which eliminates the need for expensive infrastructure and cabling (Tronci et al., 2022).
2. Flexibility: WSNs are flexible and can easily be relocated or reconfigured to adapt to changing conditions or requirements (He et al., 2019).

3. Easy deployment: WSNs can be rapidly deployed in remote or difficult-to-reach areas, which make them well-suited for monitoring and controlling various physical systems (Li & Liu, 2020).
4. Real-time monitoring: WSNs provide real-time monitoring and control, which allows for timely and effective response to changes in the environment (Abdulwahab et al., 2022).
5. Improved accuracy: WSNs can provide more accurate data than traditional monitoring methods, as they can measure multiple parameters simultaneously and at high frequency (Yu et al., 2016).
6. Scalability: WSNs can be easily scaled up or down to meet changing requirements, making them suitable for a wide range of applications (Khan et al., 2019).
7. Energy efficiency: WSNs are designed to operate with low power consumption, which helps to extend their battery life and to reduce the cost and environmental impact of battery replacement or recharging (Xu et al., 2019).

2.3.2.2 Challenges of WSNs

While WSNs offer many benefits, they also provide several challenges, including:

1. Limited energy resources: Sensor nodes in WSNs typically operate on batteries, which have limited energy resources (Kocakulak & Butun, 2017). Managing energy consumption is a key challenge in WSNs design, as energy efficient algorithms are needed to extend the life of the nodes (Ibrahim & Zaidan, 2021).
2. Interference and noise: WSNs operate in a noisy and dynamic environment, which can result in interference and reduced communication range (Kulkarni et

al., 2020). This can make it difficult to ensure reliable and efficient communication between nodes.

3. Security: WSNs are vulnerable to security threats such as hacking, spoofing, and eavesdropping, as they often operate in unsecured environments and transmit sensitive data (Khattak, et al., 2019). Ensuring the security of WSNs is a critical challenge (Bangash & Al-Salhi, 2017).
4. Scalability: WSNs can be difficult to scale up: adding more nodes to the network can result in increased energy consumption and communication overheads (Jan et al., 2017).
5. Reliability: WSNs must be highly reliable: failures in the network can have serious consequences, such as loss of data or incorrect control actions (Deif & Gadallah, 2017).
6. Heterogeneity: WSNs can be heterogeneous, with nodes having different processing, storage, and communication capabilities (Sharma et al., 2019). Managing this heterogeneity can be a challenge: different nodes may require different protocols or algorithms to ensure efficient and effective communication (Bhushan & Sahoo, 2018).
7. Data processing: WSNs generate large amounts of data, which can be difficult to process and analyse in real-time (Syafrudin et al., 2018). Efficient in-network data processing and data aggregation algorithms are needed to minimize the amount of data transmitted to the central location (Kiruthiga & Shanmugasundaram, 2021).

The main focus of this research is to investigate the security challenges faced by WSNs due to their limited energy resources, with a specific emphasis on the Denial of Sleep (DoSL attack).

2.4 Denial of Sleep (DoSL) Attack

The DoSL is a denial-of-service (DoS) attack that targets the ability of a device or system to enter into a low-power sleep mode (Khatkar et al., 2020). The attack interferes with the normal functioning of the device or system by continuously sending it data or requests that keep it active, preventing it from entering into a low-power state (Lee et al., 2020). This type of attack is particularly relevant for battery-powered devices and systems, as it can quickly drain their batteries and render them useless (Scaife et al., 2018). It can also strain the power grid and increase energy consumption, leading to higher energy costs and potential environmental impacts (Reyna & Chester, 2017).

DoSL attacks can have a number of negative consequences, including:

1. Draining of batteries: The device or system is prevented from entering into a low-power state, leading to a rapid depletion of its battery life (Goknil & Yildirim, 2022).
2. Performance degradation: The device or system is unable to enter into a low-power state to conserve energy and perform maintenance tasks, leading to decreased performance and increased latency (Razawt et al., 2017).
3. Increased energy consumption: The device or system is continuously active, leading to increased energy consumption and strain on the power grid (Kuang et al., 2017).
4. User experience compromise: The attack can negatively impact the user experience by causing the device or system to become unavailable or unresponsive (Xenofontos et al., 2017).

DoSL attacks can be classified into different groups according to their level of destructiveness, such as:

1. Attacks that disrupt the flow of information within the network.
2. Attacks that cause WSNs to be slow or low in functioning.

A transmitter uses scientific sensors such as thermocouples; flow meters, etc., to measure a specific property in a process. Thereafter it will transmit data via radio signals to a receiver. The receivers then receive and interpret the wireless data. It is not uncommon for sensor nodes to run out of resources. (Therefore, it will be embedded in WSNs from outside or the node could be a legitimate node that has been compromised). Some DoSL attack indicators are listed below (Gavric & Simic, 2018):

- Attacks increasing spam messages
- Attacks on the network that cause it to slow down
- Attacks that result in some parts of the network becoming unresponsive.

In this research, the focus is on the impact of DoSL attacks on energy consumption.

DoSL attacks in WSNs are security threat that targets the power supply of sensor nodes, resulting in their exhaustion and subsequent inability to perform their intended functions.

There are several algorithms that have been proposed to detect and defend against DoSL attacks in WSNs. Some of the popular ones include:

1. The energy-based detection algorithm: This algorithm monitors the energy consumption of each sensor node and compares it with a predefined threshold (Alagirisamy & Chow et al., 2018). If the energy consumption exceeds the

threshold, this is an indication of a DoSL attack; and the node is isolated (Biron et al., 2018).

2. The distance-based detection algorithm: This algorithm calculates the distance between the attacking node and its neighbours, comparing it with a predefined threshold (Alajlan et al., 2022). If the distance is greater than the threshold, this is an indication of a DoSL attack; and the node is isolated (Zou et al., 2019).

3. The reputation-based detection algorithm: This algorithm assigns a reputation value to each sensor node based on its behaviour and communication history (Luo et al., 2020). Nodes with a low reputation value are more likely to be involved in a DoSL attack; and are therefore isolated (Zheng et al., 2021).

4. The sleep scheduling algorithm: This algorithm schedules the sleep and waking periods of sensor nodes to balance the energy consumption and prolong their lifespan (Sinde et al., 2020). The algorithm reduces the impact of DoSL attacks and ensures the longevity of the network (Singh & Saini et al., 2021).

These algorithms can be used individually or in combination to enhance the security of WSNs against DoSL attacks (Bhola et al., 2019). However, it is important to note that no single algorithm can provide complete protection against DoSL attacks; and that a multi-layered approach is necessary for a robust defence.

2.5 Related Work

Numerous studies have developed and employed a variety of security measure in addressing DoS attacks and to secure WSNs. This is not a new problem: it was formerly widely investigated. Various methods were provided by researchers to cope with such attacks: most of the existing models were mainly concerned about energy consumption. It is pertinent to note that in the context of DoSL, a number of

approaches exist to curb these attacks; however, the majority of such are techniques that do not take energy efficiency into consideration. Even when they do, throughput becomes a trade-off which could become counter-productive in the long run.

Pawar et al. (2018) proposed a novel MAC-layer mechanism: the green and secure hybrid medium access control (GSHMAC) algorithm, to counteract DoSL in WSNs. The proposed mechanism is a cluster-based hybrid MAC mechanism that uses a mix of time-division multiple accesses (TDMA) and carrier-sense multiple access (CSMA) modes for intra-cluster communication; and a TDMA mode only for the inter-cluster communication used. During intra-cluster communication, the MAC mechanism shifts the mode according to a collision threshold. The GSHMAC is inherently safe; it implements cluster-based networks, in which an attack in one cluster cannot easily penetrate another cluster. The other security perspective in the GSHMAC is the use of the TDMA mode for inter-cluster communication; and a mixed TDMA and CSMA mode for intra-cluster communication. When a node switches to a TDMA mode, it only communicates during allocated slots, and therefore an attack will only be successful if malicious nodes manage to steal those slots. In addition, the GSHMAC modifies the internal MAC mechanism (mode-shift mechanism of hybrid MACs according to the traffic conditions and the cluster-based topology used) to further improve security. In this regard, the use of the collision threshold to detect collision attacks is introduced along with the maintenance and analysis of a retry counter to detect replay attacks. In addition, the combination of the two countermeasures improves security against a full-domination attack. It is claimed that simulation results using NS-2 of the GSHMAC scheme ensure better improvement in energy consumption and throughput. However, the scheme does not bring significant improvements in the end-to-end delay network. The reason for such a trade-off is the time required for mode switching and TDMA

scheduling in a hybrid MAC scheme. On the other hand, our algorithm considered minimizing end-to-end delay in WSNs. Our algorithm is similar to the one proposed in this study. But considers using a hybrid MAC scheme for DoSL that uses an internal MAC mechanism while improving power consumption and throughput. Moreover, our algorithm uses a hybrid approach.

Fotohi & Bari (2020) proposed a novel countermeasure technique to protect WSNs against DoSL attacks using Firefly and Hopfield neural network (HNN) algorithms. Thus, the mobile sink is applied to both improve energy consumption and increase network lifetime. The Firefly algorithm is proposed to cluster nodes and authenticate on two levels to prevent DoSL. In addition, the Hopfield neural network detects the direct route of the sink movement to send data to the cluster head (CH). Furthermore, the WSN-FAHN technique is assessed through wide simulations performed in the NS-2 environment. The WSN-FAHN procedure superiority is demonstrated by simulation outcomes in comparison with contemporary schemes based on performance metrics. The authors claim that their scheme decreases energy consumption and ensures an average throughput. However, our proposed algorithm does take into account the end-to-end delay, which was not considered in the previous algorithm.

Gunasekaran & Periakaruppan (2017) presented the generic algorithm-denial-of-sleep attack detection (GA-DoSLD) algorithm for analysing the misbehaviour of the nodes. The algorithm implements a Rivest Shamir Adleman (RSA) algorithm in the base station (BS) for generating and distributing the key pair among the sensor nodes. Before sending/receiving the packets, the sensor nodes determine the optimal route using an ad hoc on-demand distance vector routing (AODV) protocol; ensuring the trustworthiness of the relay node using the fitness calculation. The crossover and mutation operations detect and analyse the methods that the attackers use for

implementing the attack. On determining an attacker node, the BS broadcasts the blocked information to all the other sensor nodes in the network. The authors claimed that by exploiting the proposed GA-DoSLD algorithm, the attacker nodes that introduce the DoSL attacks are removed from the communication, thus saving the energy consumption of the sensor nodes. However, the AODV protocol suffered in optimizing energy consumption. In this study, the S-MAC protocol is used on the basis of the periodic listen-sleep mechanism of nodes for avoiding energy consumption. The proposed algorithm also used a modified RSA algorithm base cryptography to prevent attacks and to perform the key exchange. Moreover, the proposed algorithm employed the advanced encryption standard (AES) to protect keys against attacks. The proposed scheme is a hybrid method called dysfunction sensor detection-RSA (DSD-RSA); and the network is divided into clusters. This phase is performed using an energy- (and distance-) based approach to select the appropriate cluster head, thus improving the network and optimizing energy consumption.

Fouchal et al. (2018) presented novel approaches for detecting DoSL attacks in cluster-based sensor networks. This method is based on the election of controller nodes which observe and report DoSL attack activities. The role of a control node is to analyse traffic and to return a warning to the cluster head should any abnormal traffic be detected. The proposed dynamic solution claimed to improve the network lifetime by minimizing the energy consumption for each sensor node; such can improve security by preventing attacks. Simulation results prove that the presented algorithm is optimal compared with the existing algorithms such as X-MAC, ZKP, and TE2P schemes. Our algorithm uses the same scenario which is to minimize energy consumption. However, security in WSN was not addressed in cluster-based sensor networks. In DSD-RSA algorithm security is taken into account. We present a low-

power cyber-security mechanism for WSNs-based smart grid monitoring applications. Our mechanism can detect and isolate various attacks such as the denial of sleep, forge, and replay attacks in an energy-efficient way. The simulation results show that our mechanism can outperform existing techniques in power efficiency while maintaining constant delay and reliability values.

Suryaprabha & Kumar (2020) proposed an optimized DoS detection algorithm to enhance the security of WSNs. The proposed algorithm involves using a combination of machine learning and statistical analysis to detect DoSL attacks in the network. The algorithm is optimized to minimize energy consumption by reducing the number of false positives and false negatives. The proposed algorithm uses a combination of a decision tree algorithm and statistical analysis to detect DoSL attacks. The simulation results using MATLAB show that the proposed algorithm is effective in detecting DoSL attacks and in reducing energy consumption. The proposed algorithm achieves a higher detection rate and a lower false positive rate compared with the existing solutions. However, the proposed algorithm failed to consider the end-to-end delay in the network. This leads to a delay in delivering the data from the source node to the destination node.

Patil & Chaudhari (2016) proposed a prevention technique to address DoSL attacks in WSNs. The proposed technique involves using a reputation-based system to identify and isolate the malicious nodes responsible for the DoSL attack. The reputation of each node is calculated based on its behaviour in the network. The nodes with a low reputation score are identified as malicious and are isolated from the network. The simulation results show that the proposed technique is effective in preventing DoSL attacks in WSNs. However, the proposed technique has some limitations. It does not consider the end-to-end delay and throughput, which are

important metrics for evaluating the performance of a WSN. Additionally, the proposed technique may not be effective against sophisticated attacks that can evade reputation-based systems.

Jinhui et al. (2018) proposed an intrusion detection system (IDS) to detect hybrid DoSL attacks in WSNs using energy trust. The proposed system consists of three modules: the pre-processing module, the trust module, and the detection module. The pre-processing module filters and pre-processes the incoming data from the sensors. The trust module uses a trust model to evaluate the energy level and behaviour of the nodes in the network. The detection module detects the hybrid DoSL attacks based on the trust scores of the nodes. The IDS was evaluated using the NS-2 simulator. The results showed that the proposed system is effective in detecting hybrid DoSL attacks with high accuracy and low false positive rates. The system was also shown to have a low overhead and low energy consumption. However, the proposed system has some limitations, such as the assumption of a fixed energy level for the nodes, which may not be realistic in a practical scenario. Additionally, the system may not be effective against attacks that can compromise multiple nodes simultaneously.

2.6 Chapter Summary

This chapter evaluated the existing authentication schemes which were used to protect and secure the WSNs from DoSL attacks. Developing a mechanism that prevents WSNs from DoSL attacks and making sure the system works well, securing the correct information, is the priority. Researchers have attempted to discover and propose a security scheme that may be sufficient to secure the WSNs. The marking probability for a node in a safe network is high. As a result, when the network is not in a safe state, there is a low likelihood of a node being marked. The protection of the data collected from WSNs, either while stored inside the body sensor network, or

during their transmission outside of the WSNs, is a major unsolved concern. This shows how important it is to secure resource constraints of WSNs devices, resulting in a high demand for both security and authentication.

Chapter 3: System Design and Architecture

3.1 Introduction

The main aim of this chapter is to develop a mechanism that will protect WSNs from DoSL attacks. The chapter will discuss the security schemes integrated into this research study for increasing security robustness when developing the proposed algorithm. WSNs are gaining market acceptance in numerous sectors due to their ubiquitous nature, increasing ease of deployment, and wide range of potential applications (Rashid & Rehmani, 2016). WSNs can be deployed in unattended environments, in which a registered user can log in to the network and access data collected by the sensors. DoSL attacks in resource-constrained environments are one of the major security concerns. Given that sensor nodes have limited resources and computational power, it is desirable that the developed mechanism securing WSNs from DoSL attacks be simple and clear. At present, the IEEE802.15.4 and IEEE 802.15.4a are the only well-accepted physical and network access standards for low-power networks (Fatani, 2021).

Wong et al. (2018) proposed a password-based dynamic user authentication scheme for WSNs, which imposes a light computational load as it requires only use of a one-way hash function and exclusive XOR operations. Daset et al. (2010) identified that Wong et al.'s protocol is easily affected by DoSL attacks. In this protocol, two factors (e.g., password and smart card) are needed to prove the communicating parties. DoSL attacks and others forms of attacks are targeting this protocol, which makes it imperative to develop a mechanism that will greatly improve the security of WSNs.

In this chapter, we present the design of the proposed DSD-RSA algorithm and associated presumptions. In Section 3.2 we provide the proposed typical DSD-RSA system architecture; in Section 3.3, we provide the design of the proposed DSD-RSA

algorithm by showing its system model and pseudo code. Finally, in Section 3.4 we provide a chapter summary.

3.2 System Architecture

WSNs in the proposed DSD-RSA consist of nodes randomly distributed. In Figure 3.1, the proposed DSD-RSA architecture shows the locations of the base stations and the nodes.

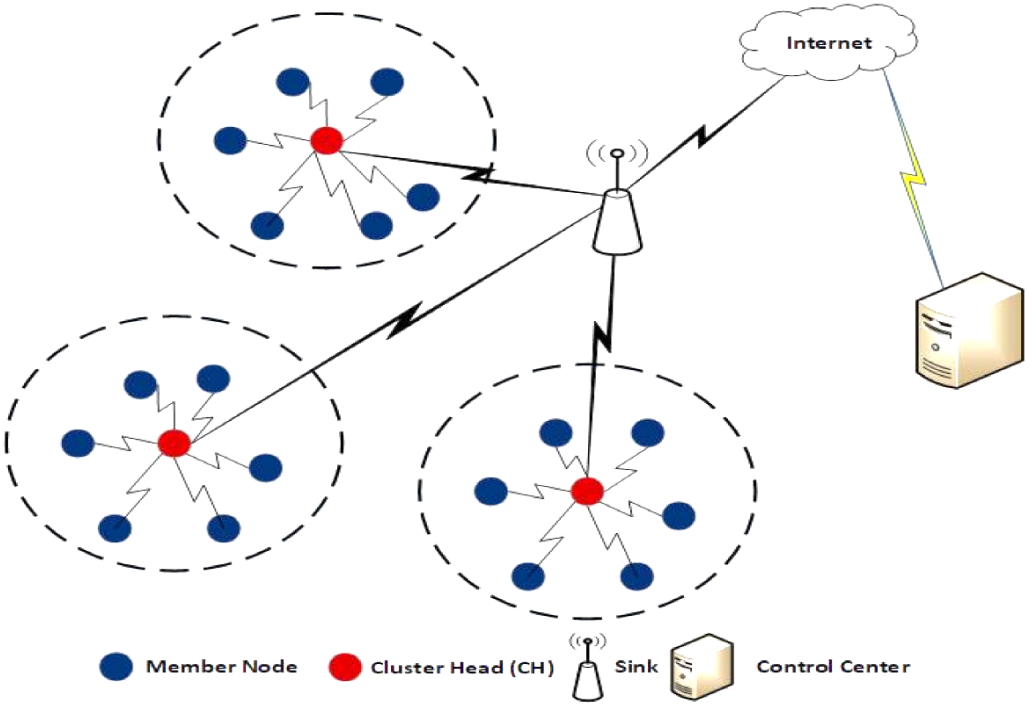


Figure 3.1: DoSL WNS systems architecture

We consider the following specifications for our proposed network:

1. All nodes are fixed; the base station receives information from the head cluster.
2. The BS is connected to the internet.
3. All nodes are homogeneous and are bound or restricted in energy.
4. The nodes are not aware of their positions; however, each node can calculate its own distance from another node according to the received message signal strength.
5. The nodes continuously sense the surrounding environment and send the relevant data at a constant rate.

3.3 Enhanced Security Prevention DSD-RSA Algorithm Design

We integrated existing models, namely, the S-MAC protocol, the AES algorithm, and the RSA algorithm when designing the proposed DSD-RSA algorithm. This was to minimize energy consumption and end-to-end delay while maximizing network throughput in WSNs.

3.3.1 S-MAC protocol

The S-MAC protocol is designed on the basis of the periodic listen-sleep mechanism of nodes for avoiding energy wastage caused by idle listening. The S-MAC reduces energy consumption because of collision, overhearing, control-packet overheads, and idle listening. In this study, the S-MAC is introduced into the proposed DSD-RSA scheme in order to reduce energy consumption while preventing DoSL attacks. Every node in the S-MAC protocol has a listen-and-sleep schedule in order to reduce energy consumption (Gunasekaran & Periakaruppan, 2017).

In sensor networks, the S-MAC protocols set the sleep periods. In these protocols, synchronization signals are transmitted to adjust the node clocks. A large number of control messages, such as RTS and CTS, are employed, which are referred to as synchronization packages (SYN). An effective technique in performing DoSL attacks is repeating control packets such as RTS messages, to prevent the nodes from going into sleep mode, resulting in waste of energy. A transmission stream of such messages with short intervals will not give the network nodes sufficient idle-state time to initiate a sleep mode, thus leading to battery power loss. While the attack is ongoing, all nodes can be affected by this damage, every node in the WSN having certain energy levels and possessing certain geographic coordinates. Regarding the characteristics of such networks and the significance of their energy, if a network node can be selected as a cluster head (CH) to reduce the required space compared with the sink or base and consuming more energy, the network would have a longer life span. To aggregate the data on a given network, it is better to analyse the data individually in each node and to transmit the results to the CH in order to reduce the computational burden for each CH. In WSNs, if the node count is less than five, clustering is not mandatory; and data are transmitted directly to the sink.

In the proposed DSD-RSA algorithm, the appropriate CH for collecting data is selected initially. The steps in selecting the CH are as follows:

Step 1: First, it is best to select a node closer to the base station. A Euclidean equation (3.1) is employed for this. The nodes in the network are all assumed to be equipped with a global positioning system (GPS), and are aware of their geographic location.

$$CS = \sqrt{(x_i - x_s)^2 + (x_i - x_s)^2} \quad (3.1)$$

where C is the coordinates of each node, and S is the sink coordinate.

Step 2: Next, the more energy-efficient of the selected nodes, and those closest to the base station are more suitable for the CHs. According to Equation (3.2), the energy of the nodes is measured as follows:

$$Q = \frac{e}{t} \quad (3.2)$$

Where e represents the node initial energy in kilojoules, while t is the time parameter in minutes. Equation (3.2) can be rewritten by converting the values in terms of jules and seconds (in the first round all nodes have homogeneous and identical energies).

Step 3: After calculating the energy for the nodes, and determining the nodes and the base-station distance, Equation (3.3) is employed to select the CH.

$$n_i = e_i . d(CS) \quad (3.3)$$

In Equation (3.3), e_i represents the energy of the nodes, while d (CS) represents nodes and the sink distance. Each node transmits its calculated energy value to the neighbouring nodes closest to the sink. Each node compares the n_i of its neighbouring

node with its own. If their n_i is lower than their neighbours, the node with the lower energy is a normal node.

Step 4: To select the cluster head, the energy consumed in each node should also be calculated in the fourth step. We could thus opt for the node with the maximum available energy as a cluster head. For this purpose, we use Equation (3.4):

$$e = Q.t \tag{3.4}$$

Finally, the nodes with the most remaining energy and the least distance to the sink are selected to be cluster-head nodes. The cluster-head selection is such that all nodes send off their remaining energy and distance to the nodes near the sink. Nodes with the most energy introduce themselves as the cluster heads, informing their cluster members by sending a message. In the next section, a process is discussed to prevent DoSL attacks.

In addition, to demonstrate detection and preventing of DoSL attacks, the DSD-RSA is implemented on an S-MAC protocol. To prevent such attacks, we use the RSA algorithm and an interlock protocol to perform key exchange, plus a method for node authentication. The procedure for transferring the keys to other nodes is a vital process, keys being prone to attacks.

Major energy consumption in WSNs is necessary to perform sensing, processing of the data, and communication. In this research study, the first-order ratio model is utilized as the energy consumption model, and the energy consumption is considered for data communication only. Relative to the sender and receiver nodes' distance (*di*), energy consumption will be modelled as a free space or a multipath fading channel.

In Equation (3.5), the amount of energy consumed to transmit *T* bits of data to distance *di* is demonstrated:

$$E_{tx}(T,di)=\begin{Bmatrix} T.E_{elec}+T_{efs}.\ d^2 \quad di,<di_0 \\ T.E_{elec}+Ke_{mp}.d^4\ di\geq di_0 \end{Bmatrix} \tag{3.5}$$

Where E_{elec} is the amount of energy required to activate the circuits; E_{mpi} and E_{fsi} are the relay activation energy model for multi-path channels and empty space, respectively; di^2 denotes the power loss in free space, while power loss in multipath fading is di^4. Currently, there is a great deal of research being conducted in the area of low-energy radios. In this research, we assume a simple mode in which the radio dissipates $E_{elec}=100nj/bitin$ the transmit or receive circuitry; and $E_{mpi=}0.0015pJ/bit/m^4$ for the transmit amplifier.

3.3.2 RSA algorithm

This study adopts the RSA algorithm for key distribution and decrypting of the message. Encryption serves as a mechanism to ensure confidentiality. Since digital data may be visible on the internet, sensitive information such as passwords and personal communication may be exposed to potential interceptors. To prevent DoSL, the RSA algorithm performs key exchange, plus a method for node authentication.

The attacker transmits SYNC messages, indicating the next sleep interval for the nodes. When a SYNC packet is received with similar sleep timing from another node,

the upcoming sleep time is calculated, to maintain the synchronization. Equation (3.6) was used to calculate the New SleepTime value.

$$New_{SleepTime} = \left(Old_{sleep\ Time} + \frac{\text{Received SYNPacket Sleep Time}}{2}\right) \quad (3.6)$$

According to this method, sleep timing is not affected abruptly with the receipt of an SYNC message. In fact, this method enables gradual synchronization of the nodes, with a similar programme. The main reason is that the attacker, who is monitoring the network, can easily identify these packages even should they be encrypted. To do this, the attacker should merely monitor all messages for their size and time. For instance, the size of S-MAC SYNC packets is ten bytes.

It is important to note that sleep modes allow the user to pause or turn off the unused modules in the node, significantly reducing energy consumption, resulting in a decrease in end-to-end delay and an increase in network throughput. Therefore, the overall energy consumption is reduced in general. In total, the proposed DSD-RSA scheme can improve QoS parameters such as delay and network throughput.

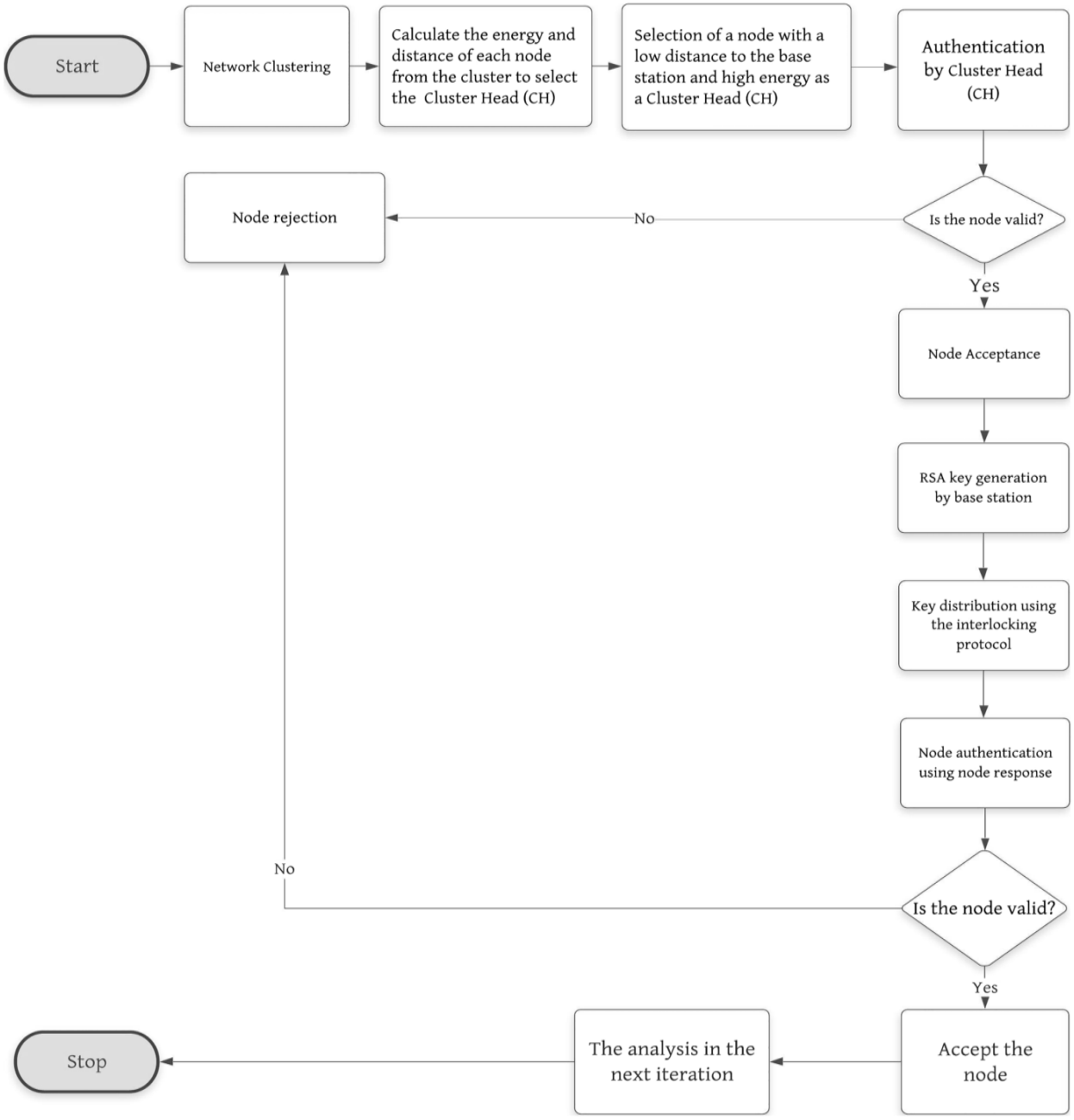
3.3.3 AES algorithm

The AES algorithm is a symmetric encryption algorithm and an interlocking protocol that uses the same key for encryption and decryption. An interlocking protocol is a protocol designed to frustrate eavesdropper attacks against two parties that use an anonymous key exchange protocol to secure their conversation. AES encrypts data in a 128bit block and uses 128bit, 192bit, or 256bit keys. This research introduced the AES algorithm to encrypt digital data in the case of the key management being vulnerable; or should the key be stolen and accessed. The AES algorithm is implemented in the proposed model to perform another cryptic operation using the cipher-text that was encrypted through the RSA algorithm via a 256-bit key with 14

rounds. The encryption is to be carried out using the AES algorithm. The AES is the management of the encryption key. To protect the keys against such attacks, a key-transfer operation is performed using an interlocking protocol, in which the AES algorithm is employed to perform key encryption (Farooq, 2017).

3.4 Enhanced Security Prevention DSD-RSA Algorithm Flowchart.

In this section, we provided a flowchart of the proposed DSD-RSA algorithm.



Algorithm 3.1 DSD-RSA algorithm flowchart

Next, in Section 3.4.1, we present the developed DSD-RSA algorithm.

3.4.1 Enhanced security-prevention DSD-RSA algorithm

We integrated the S-MAC protocol, AES, and RSA algorithms when designing the proposed DSD-RSA algorithm with the aim of detecting and preventing DoSL attacks, while satisfying the QoS in terms of energy consumption, end-to-end delay, and throughput, as given by Algorithm 3.1.

Algorithm 3.1 Pseudo code for DSD-RSA proposed model

1. Step 1. Network clustering;
2. Create the random WSN as sensor nodes;
3: Define parameters of Algorithm1;
4: Select the cluster head (CH);
5: Calculate the energy of each node;
6: Calculate the distance of each node;
7: Select a node as a CH with a low distance from the base station and high energy;
8. Procedure node authentication.
9: Step 2. Node authentication at cluster level;
10. Calculate threshold;
11. Calculate interval;
12. While interval > threshold do;
13. For all of the node at cluster level;
14. Should be done authentication phase;
15. End for;
16. Else;
17. Node is accepted;
18. End while.

19. Step 3. General authentication in the network;
20. RSA key generation by base station (BS);
21. Node authentication is conducted using node reply;
22. If the node is invalid
23. the node is rejected;
24. The node is accepted;
25. End if;
26. End procedure.

Algorithm 3.1 DSD-RSA algorithm

We generate public and private keys for each sensor node's communication, so that they can be provisioned in the network and authenticated as valid nodes. The RSA key-generation algorithm is used for this purpose. The RSA is used because, with symmetric key encryption algorithms, both the sender and the recipient should be aware of the encryption key. One of the recipients of the message can jeopardize the security of everyone if he discloses a key the sender has used to encrypt the message. The RSA algorithm is a public key method. This method is the first trusted method among other encryption methods, and is one of the biggest enhancements of cryptography. In public-key algorithms, two completely different keys are used for encryption and decryption: the public key and the private key. The public key is used to encrypt information. Everyone knows it, because this key is used only to encrypt the information. Enemies will not be able to decrypt the data encrypted by others. The private key is the key used to decrypt the encrypted data. This key is not even known to friends and trustees. Thus, any network-level entity (whether user, machine, or process) requires two independent keys, only one of which is sensitive and must be

kept secret. The nature of this cryptographic algorithm is such that in practice the private key cannot be deduced from the public key. Public and private keys can be exchanged using the interlocking protocol to protect these communications against attacks (Sicari et al., 2018; Gavric & Simic 2018; Premkumar & Sundararajan, 2020).

3.5 Chapter Summary

This chapter presented the design of the enhanced security DoSL attack by integrating the S-MAC protocol, AES, and RSA algorithms in order to provide an accurate network that secures WNS over DoSL by reducing energy consumption while ensuring QoS in terms of end-to-end delay and throughput. When designing the proposed DSD-RSA algorithm, we used the S-MAC for sensor networks that have sufficient energy. The major applications of S-MAC include tolerance delay and long idle listening. RSA algorithm-based cryptography was utilised to prevent attacks and to perform key exchanges; while the AES algorithm was employed to protect keys against attacks. The sleep nodes allow the user to pause or turn off the unused modules in the node, significantly reducing energy consumption, resulting in a decrease in end-to-end delay and an increase in network throughput.

Chapter 4: System Implementation

4.1 Introduction

In this chapter, we provide an insight into the implementation of the proposed DSD-RSA algorithm. We also discuss an overview of various network simulators and provide the comparison between these simulators. However, we only discuss the open-source and platform-independent network simulators. In this work, the simulations were carried out using NS-2 version 2.35. Therefore, we also discuss the reason for choosing to use NS-2 rather than other network simulators. We ran our NS-2 on Linux Ubuntu 14.04 64-bit, 1 GB RAM, and 10 Gigabyte (GB) of storage. The operating system (OS) was installed on an Oracle Virtual Box Manager (VM) Version 6.1.30 developed in 2021 by Oracle Corporation. The simulation the proposed DSD-RSA model and network topology configurations were using was C++ source code and TCL scripts. We further discuss the NS-2 environment, which includes its basic architecture, directory structure, class hierarchy, as well as trace file output format. We also provide the steps involved in the configuration and installation of NS-2 on Linux Ubuntu-bit.

The remainder of this chapter is organized as follows: In Section 4.2, we briefly discuss the existing network simulators. In Section 4.3, we present the comparison between these network simulators. In Section 4.4, we discuss the NS-2 environment and give its advantages. In Section 4.5, we provide a detailed discussion on how we simulated and tested our model. Finally, in Section 4.6, we summarize the chapter.

4.2. Overview of Network Simulators

Currently, there are many network simulators that have different features for different aspects. A short list of the current network simulators includes NS-2, Network

Simulator-3 (NS-3), OPNET, Optical Micro-Networks (OMNeT++), REAL, SSFNet, J-Sim, and QualNet (Fakhar, 2019). A network simulator is a software programme that can predict the performance of a computer network. The primary reason for using a network simulator is that setting up a real network is expensive as well as time-consuming. The purpose is to model the behaviour of real communication networks. Network simulation is the most useful and common methodology for evaluating various network topologies without real-world implementation. In this section, we provide an overview of the most used network simulators, giving a comparison of these network simulators. However, as mentioned in Section 4.1, in this study, we will only discuss the freely distributed and open-source network simulators, which also provide free access to the available source codes, etc. We further discuss the reason for NS-2 being preferred to other simulation tools for simulation in WSNs.

4.2.1. Network Simulator-2

Network Simulator (Version 2) widely known as NS-2, is simply an event-driven simulation tool that has proved useful in studying the dynamic nature of communication networks (Jamali & Fotohi, 2017). A typical NS-2 simulation interface is illustrated by Figure 4.1. NS-2 refers to an object-oriented discrete-event network simulator targeted at networking research and originally developed in 1989 by Marc Greis.

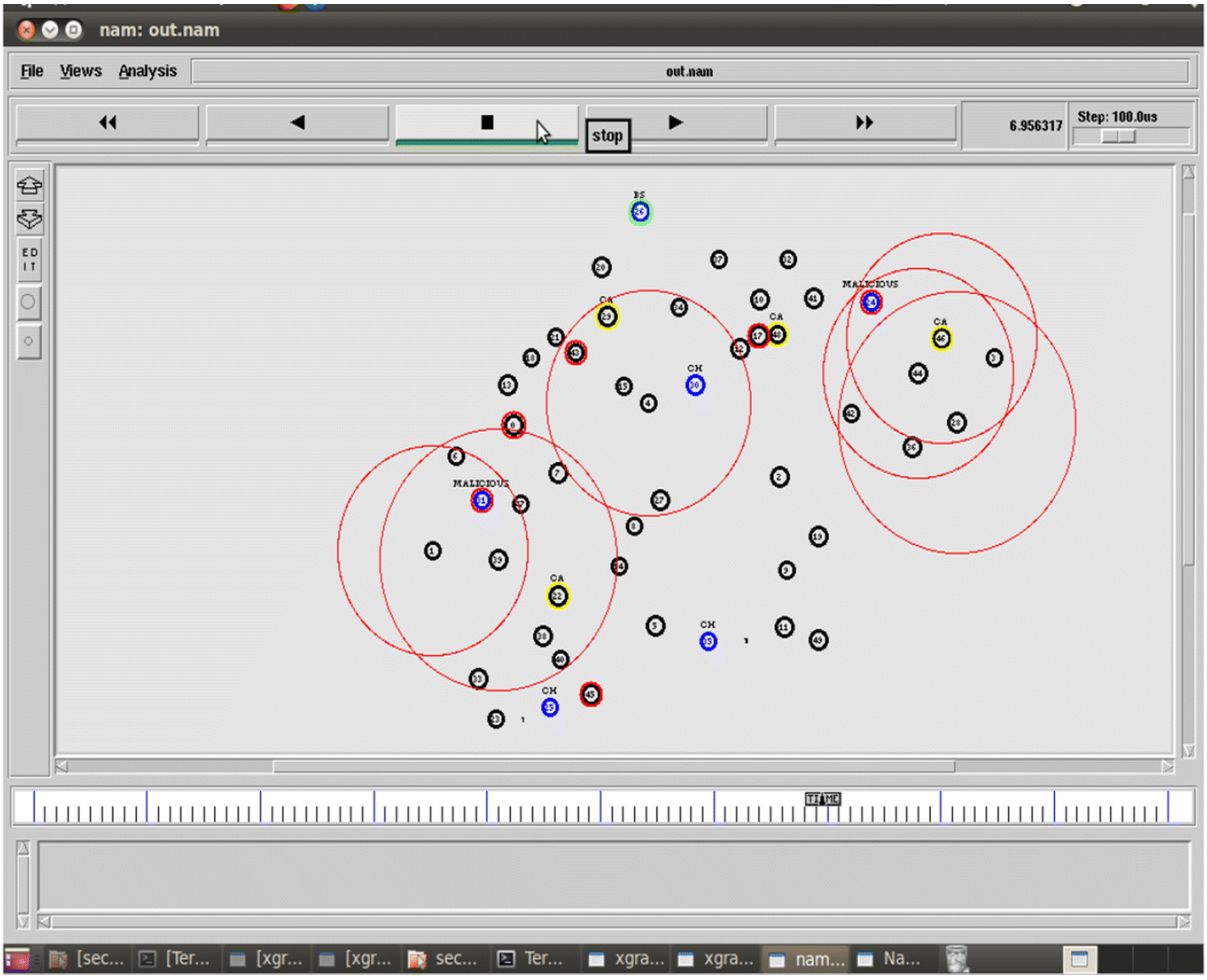


Figure 4.1: NS-2 simulation interface

Simulation of wired as well as wireless network functions and protocols (e.g., routing algorithms, TCP, UDP) can be conducted using NS-2. In general, NS-2 provides users with a way of specifying such network protocols and simulating their corresponding behaviours (Sundaram et al., 2020). Due to its flexibility and modular nature, NS-2 has, since its birth in 1989, gained constant popularity in the networking research community. Ever since, several revolutions and revisions have marked the growing maturity of the tool, thanks to substantial contributions from the players in the field (Glaropoulos et al., 2015). Among these are the University of California, U.S. state of California and Cornell University where the REAL network simulator was developed, the foundation on which NS-2 was invented (Dorathy & Chandrasekaran, 2018). Since

1995, the Defence Advanced Research Projects Agency (DARPA) has supported the development of NS-2 through the Virtual Inter Network Testbed (VINT) project. The NS-2 consists of two key languages: C++ and object-oriented tool command language (OTcl). While the C++ defines the internal mechanism, (i.e., a backend) of the simulation, the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events, (i.e., a frontend). The C++ and the OTcl are linked together using TclCL. Mapped to a C++ object, variables in the OTcl domains are sometimes referred to as handles.

NS-2 is an effective tool for reflecting how the attacks are formed as well as their preventions; the motivation for the research work being to secure the CH from denial of sleep attacks. Using the NS-2, a network is first created from which a cluster is formed. The cluster is then attacked by the outside node which is shown using TCL script. In NS-2 there are various input files which can be used for further modification of any protocol.

4.2.2. Network Simulator-3

NS-3 has been available to the public since 2006 (Henderson et al., 2006). Similar to NS-2, it is an open source or free software, licensed under the GNU GPLV 2, discrete-event network simulator, written in C++ language with some parts of the simulation written in Python. Similarly, to NS-2, NS-3 does not have an OTcl API. NS-3 is considered a replacement of NS-2, rather than an extension. As a result, translating a network topology in NS-3 is very difficult (Rampfl, 2013). The NS-3 interface is shown in Figure 4.2.

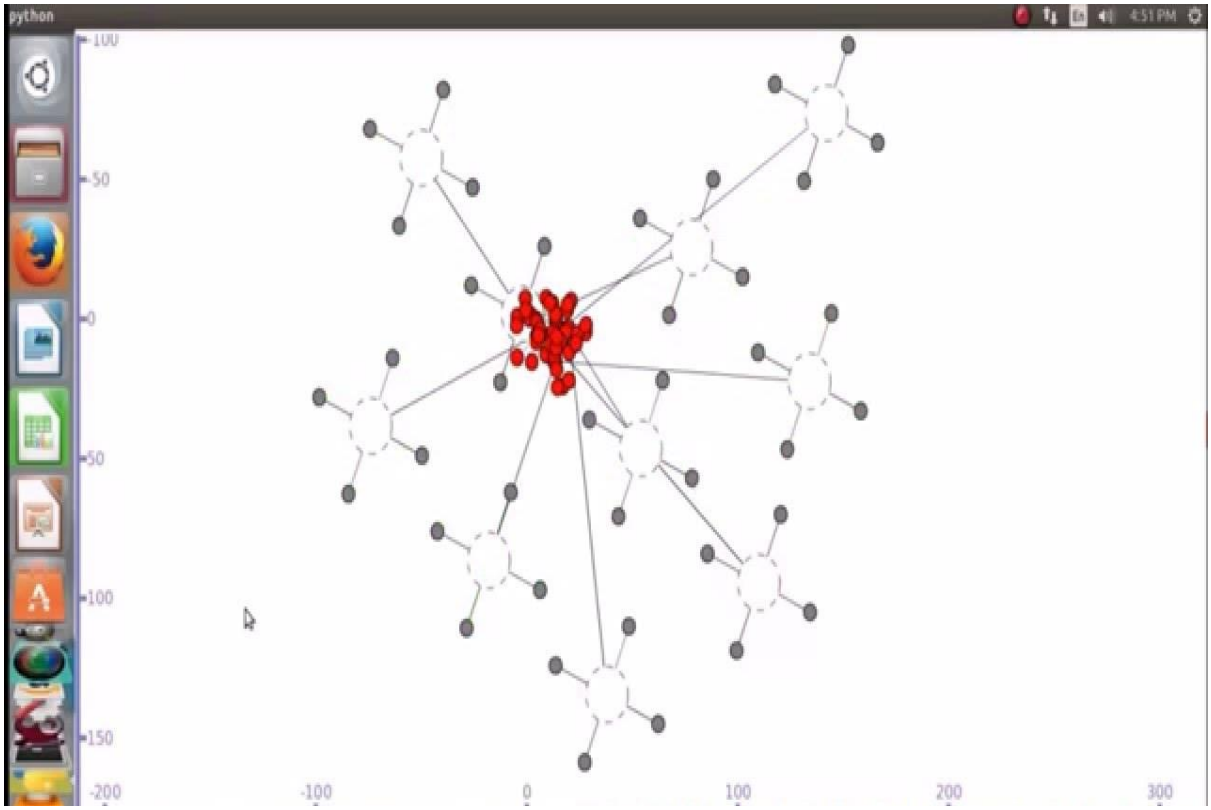


Figure 4.2: NS-3 simulator interface

4.2.3. Optical Micro-Networks ++

OMNET++ was initiated in September 1997; currently, OMNET++ has a huge number of operators (Campanile et al., 2020). OMNET++ is a discrete-event, open source, written in C++ language. Unlike NS-2 and NS-3, OMNET++ is not only used for network simulations (Manishankar et al., 2019). OMNET++ can be utilized for multiprocessor modelling and complex software systems performance evaluation. In addition, OMNET++ provides intelligence support and extensive graphical user interface (GUI). OMNET++ is accessible by both UNIX and Windows systems. However, it is most commonly utilized for computer networks simulation. The OMNET++ interface is shown in Figure 4.3. However, compared with other simulators, the simulations in OMNeT++ require more effort.

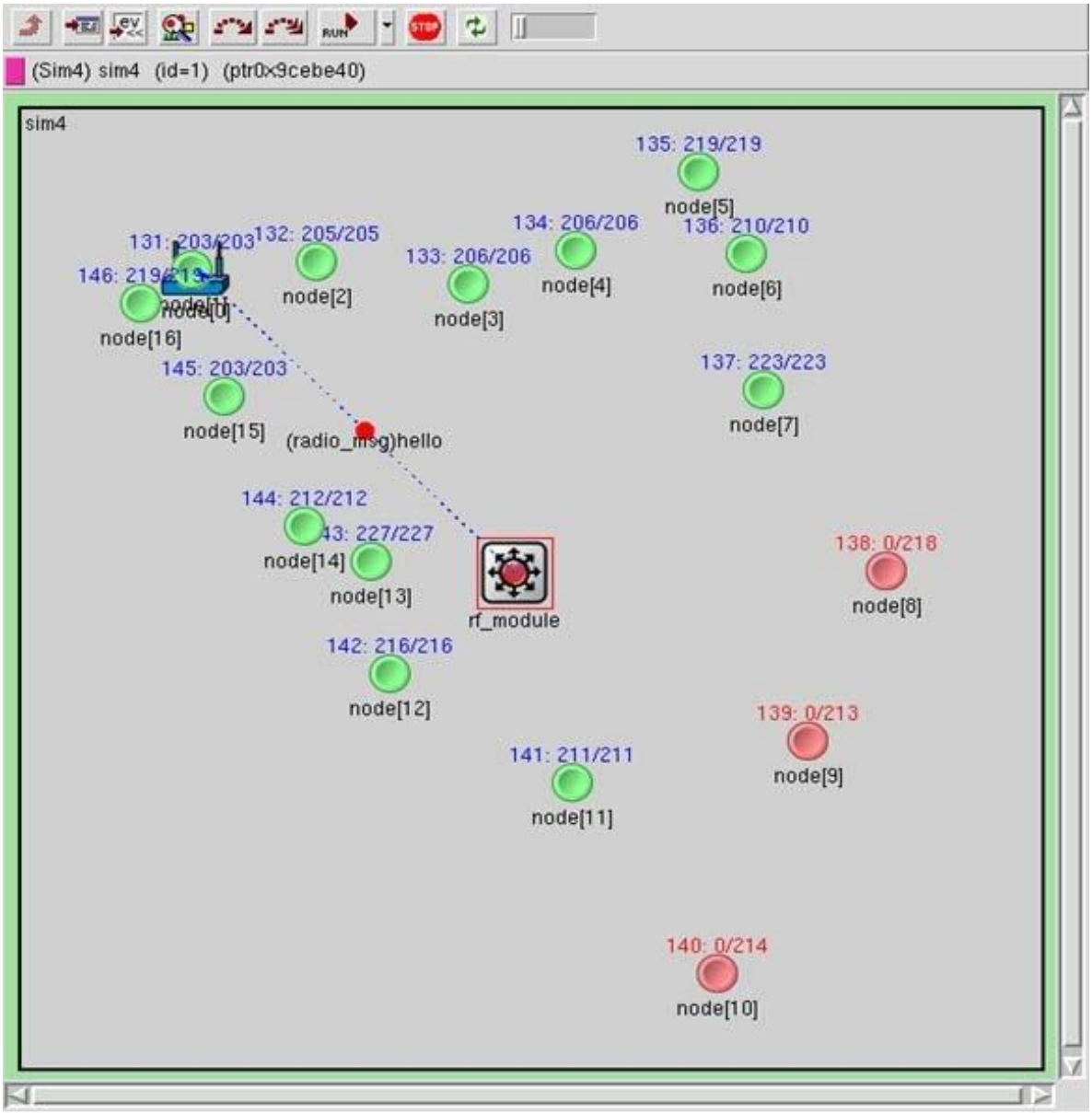


Figure 4.3: OMNET simulator interface

4.2.4. MATrixLABoratory

A typical MATrixLABoratory (MATLAB) simulation interface is illustrated in Figure 4.4. MATLAB refers to a programming language that facilitates the manipulation of the matrix, plotting of functions and data, implementation of new algorithms, as well as the improvement of existing ones; and the creation of simple user interfaces (Valentine & Hahn, 2022). MATLAB was developed in the 1970s by Cleve Moler, chairman of the Department of Computer Science at the University of New Mexico (Moler& Little, 2020). Currently, this simulation tool is maintained by MathWorks and used by many

researchers to analyse and design their systems for purposes of machine learning, communications, design control, processing of signals, computer visions, processing of images, robotics, as well as computational finances (Pierson & Gashler, 2017). In addition, this language and its code can be integrated with other programming languages such as C, C++, Java, Python, and FORTRAN. Also, it can be easily integrated with hardware models (Rajaram et al., 2016). The benefit of MATLAB is that it is platform independent. However, it is not free; and hence users must obtain a licensed version from MathWorks.

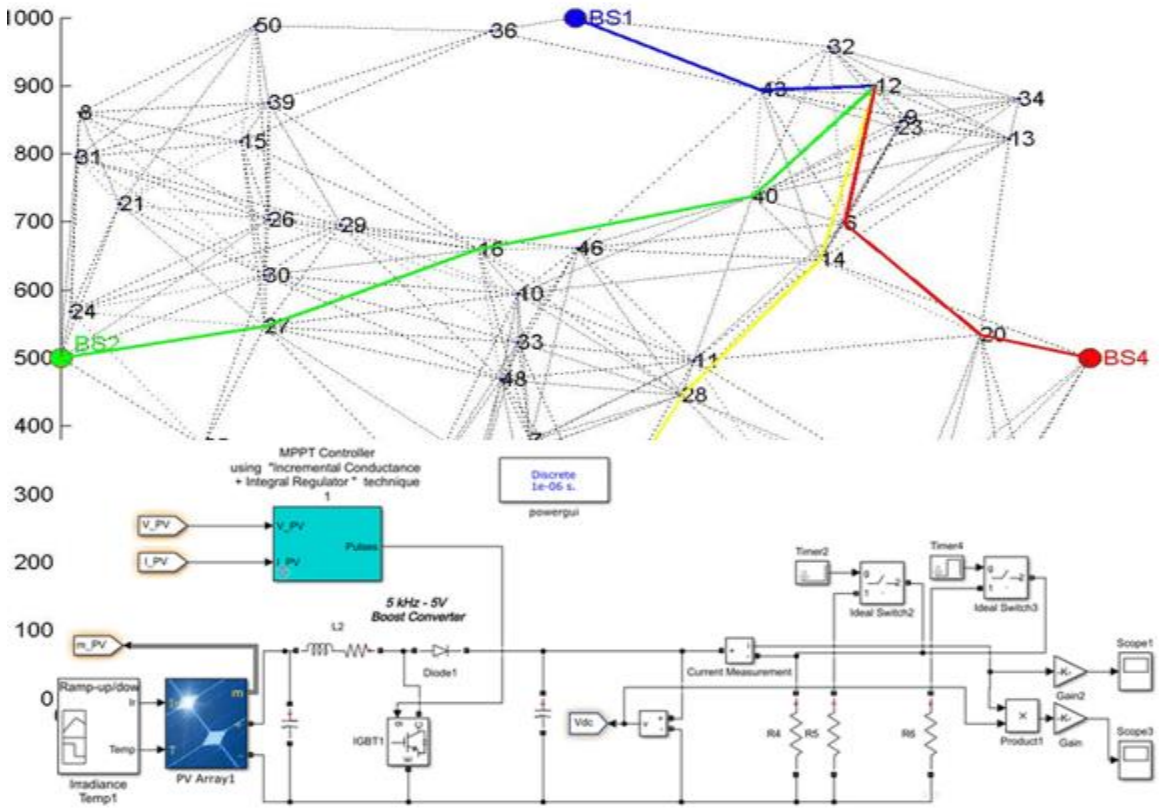


Figure 4.4: MATLAB simulation interface

4.3. Comparison of Network Simulators

The summary and characteristics of discussed network simulators are provided in Table IV. The NS-2 and NS-3 are not altogether favourable in simulating large communication networks (see Table I). However, these simulators are widely utilized by researchers all over the world, primarily because they are easy to use, unlike

MATLAB and OMNET++. All of the discussed network simulators comprise open-source code and are easily accessible on the internet. In this research, we needed to design a simple network topology, which includes mobile nodes communicating wirelessly (see Table I).

Table I: The Comparison of Network Simulators

Features	NS-2	NS-3	OMNET++	MATLAB
Ease of setup	Difficult	Difficult	Moderate	Hard
Ease of use	Easy	Difficult	Moderate	Moderate
Scalability	Poor	Poor	High	High
Large Networks	No	No	Yes	Yes
Programming Language	C++	C++	C & C++	C,C++, Java, Python and Fortran
Support Integer Linear Programme Algorithm	No	No	No	Yes
GUI	Yes	Yes	Yes	Yes
Continuous development	Yes	Yes	Yes	Yes
IEEE 802.11 protocol	Yes	Yes	Yes	Yes
Complications	No	No	No	No

Examining the characteristics mentioned in Table I, NS-2 is suitable as a simulation environment for the proposed DSD-RSA algorithm. This is mainly because it is easy

to set up, easy to operate, and it allows the IEEE 802.11 protocol to be simulated. The NS-2 is composed of C++ for core functionality of the proposed DSD-RSA algorithm. This simulator also includes a graphical user interface (GUI) for visualizing network designs and simulations. Various other tools are available for implementation of wireless sensor network simulation, such as OMNET++ and MATLAB. The NS-2 is an effective tool for showing how the attacks are formed, together with its prevention. The motivation of the research work is to secure the cluster from DoSL as depicted in Figure 4.1. Using the NS-2, a network is first created from which a cluster is formed. The cluster is then attacked by the outside node which is shown using TCL script (see Figure 4.1). In addition, in NS-2 there are various input files which can be used for further modification.

4.4 Chapter Summary

In this chapter, the research study discussed the simulation setup for the proposed DSD-RSA algorithm. The NS-2 was used as the simulation tool. The research study provided a review of open source simulators, together with their comparisons. Furthermore, the research study provided a detailed overview of the NS-2 simulator, discussing its architecture and the features that made it suitable for use in this work. In the next chapter, we present the simulation results.

Chapter 5: Results & Discussion

5.1 Introduction

Simulators have been beneficial to many researchers examining and studying applications and protocols under varying network configurations, as briefly described in Section 4.2. Simulations are used to model real-time communication behaviour of networks. Setting up a real-time communication network is time-consuming and costly. Thus, in this chapter, we discuss experimental evaluations of the DSD-RSA algorithm. Experiments were conducted based on NS-2 evaluations of three performance metrics, namely, average throughput, average delay, and energy consumption.

The reason behind these metrics is that the research has concurred that to improve QoS in any network, these performance metrics are to be considered very important. During simulations of the proposed algorithm, trace files were automatically created and used to record all the results obtained. Furthermore, those results were used to calculate average throughput, average delay, and energy consumption. In the 1970s, Cleve Moler, chairman of the department of computer science at the University of New Mexico designed and developed MATLAB. Thus MATLAB was used to display the analysed results which were obtained through extensive simulations in the form of graphs. The motive behind this language is the creation of interactive graphs.

The remainder of this chapter is organized as follows: In Section 5.2, this study presents the evaluations of the simulation. In Section 5.3, findings and discussions are provided. In Section 5.4, a summary of this chapter is presented.

5.2 Simulation Evaluation

As shown in Figure 4.2, this study used TCL scripts to create the proposed simulation topology in NS-2. This was to evaluate the proposed DSD-RSA model. During the

simulations, the NAM interface simulation topology, as explained in Section 4.2 and illustrated in Figure 4.1, was used to observe the behavior of nodes during the simulations. NAM is one of the TCL/TK-based animation tools. The tool is used to view network simulation traces and the real world (Goswami et al., 2017). The analysed results from several simulations were graphically displayed in MATLAB using the plot function and XGraph. In addition, we coded AWK programmers to calculate the total average of the performance metrics considered in this study, such as the average throughput, average delay, and energy consumption. AWK is a utility that enables programmers to write small but effective programmes in the form of statements (Talukdar, 2021). We ran NS-2 on Linux Ubuntu 14.04 64-bit, 1 GB RAM, and 10 GB of storage. The operating system (OS) was installed on an Oracle Virtual Box Manager (VM) Version 6.1.30 developed in 2021 by Oracle Corporation. The simulation of the proposed DSD-RSA algorithm and network topology configurations were implemented using C++ source code and TCL scripts. This study had extensive simulations that were performed several times to obtain the most promising and reliable results.

5.3 Findings and Discussions

The proposed model was evaluated against WSN-FAHN, GSHMAC and GA-DoSLD algorithms. The motivation behind choosing these three algorithms was that both were proposed to improve QoS in communication networking environments. Also, the focus of these algorithms was on network throughput, delay, as well as energy consumption. This study was also motivated by the following similarities and differences: the proposed DSD-RSA model integrated three algorithms, namely, S-MAC protocol, the AES algorithm, and the RSA algorithm.

We use the S-MAC protocol for sensor networks that have sufficient energy. The major applications of S-MAC protocol include tolerance delay, and long idle listening. The RSA algorithm-based cryptography was utilised to prevent attacks and to perform key exchanges; while the AES algorithm was employed to protect keys against attack. The sleep modes allow the user to pause or turn off the unused modules in the node. However, in the proposed DSD-RSA algorithm, the use of other parameters such as several types of mobile sinks cannot be guaranteed to further reduce energy consumption efficiently in wireless sensor networks, as a part of future work. This is because the evaluation was not thoroughly done.

The performance metrics defined below were considered in evaluating the performance of the proposed DSD-RSA algorithm against WSN-FAHN, GSHMAC, and GA-DoSLD algorithms.

a) **Average throughput** – the measurement of the amount of data that can be transmitted at a given time between two or more nodes within a network, calculated as: Throughput = (Data Size) / (Transmission Time).
b) **Average End-to-End Delay** – the measurement of the time it takes data to travel from one node to another within a network, calculated as: End-to-End Delay = Propagation Delay + Processing Delay + Queuing Delay.
c) **Energy consumption** – the measurement of the amount of energy consumed during data transmission within a network, calculated as: E = P * T (E is the energy consumption in joules (J), P is the average power consumption in watts (W), T is the time of operation in seconds (s)).

5.3.1 Average throughput

This average network throughput was determined using the AWK programs. The WSN-FAHN, GSHMAC, and GA-DoSLD algorithms and the proposed DSD-RSA algorithm were compared as demonstrated, promising average network throughput. In order for the algorithm to be effective, data packets must be received and transmitted accurately under various sending rates, including 1 packet per 3 seconds, 1 packet per 5 seconds, and 1 packet per 7 seconds (Hsueh et al., 2015). As a performance measure, this research utilizes a packet sending rate of 1 packet per 3 seconds. The DSD-RSA algorithm conserves energy in nodes by using a low-power sleep state when not actively transmitting or receiving data. This reduces energy consumption and increases available bandwidth for data transmission, resulting in increased network throughput. From the Figure 5.1, it is clear that the proposed DSD-RSA algorithm provides higher throughput than the existing algorithms at a packet sending rate of 1 packet per 3 seconds. The proposed DSD-RSA algorithms, GA-DoSLD, WSN-FAHN, and GSHMAC algorithms experienced an average throughput of 95%, 89%, 65%, and 60%, respectively.

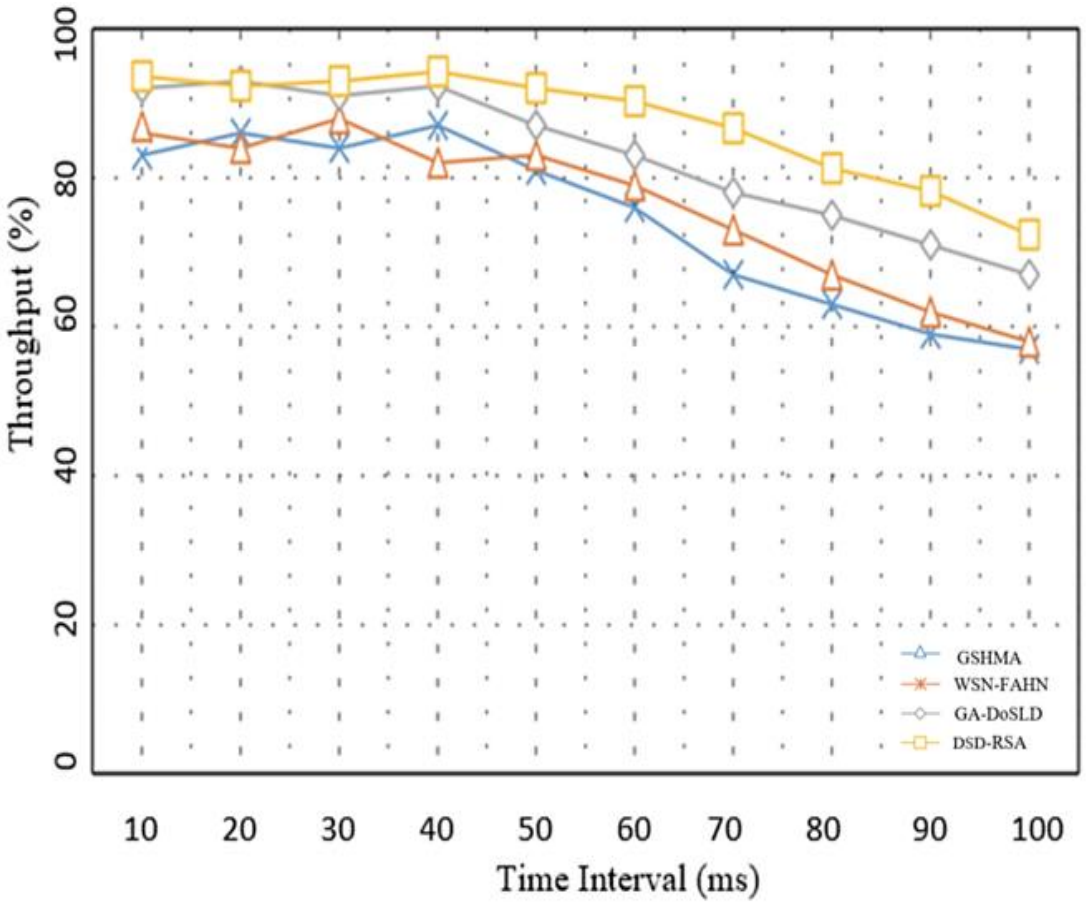


Figure 5.1: Average throughput

5.3.2 Average end-to-end delay

The end-to-end delay is defined as the average time consumed for transmitting the packets. The analysis of end-to-end delay with respect to the packet size is represented in Figure 5.2. The figure shows that the proposed DSD-RSA algorithm achieves minimal end-to-end delay for variable packet sizes by avoiding unnecessary transmissions, leading to reduced delay. By comparison with the existing GA-DoSLD, WSN-FAHN, and GSHMAC algorithms, the DSD-RSA algorithm significantly minimizes end-to-end delay. The proposed DSD-RSA, GA-DoSLD, WSN-FAHN and GSHMAC algorithms experienced an average end-to-end delay of 19%, 35%, 40% and 60%, respectively.

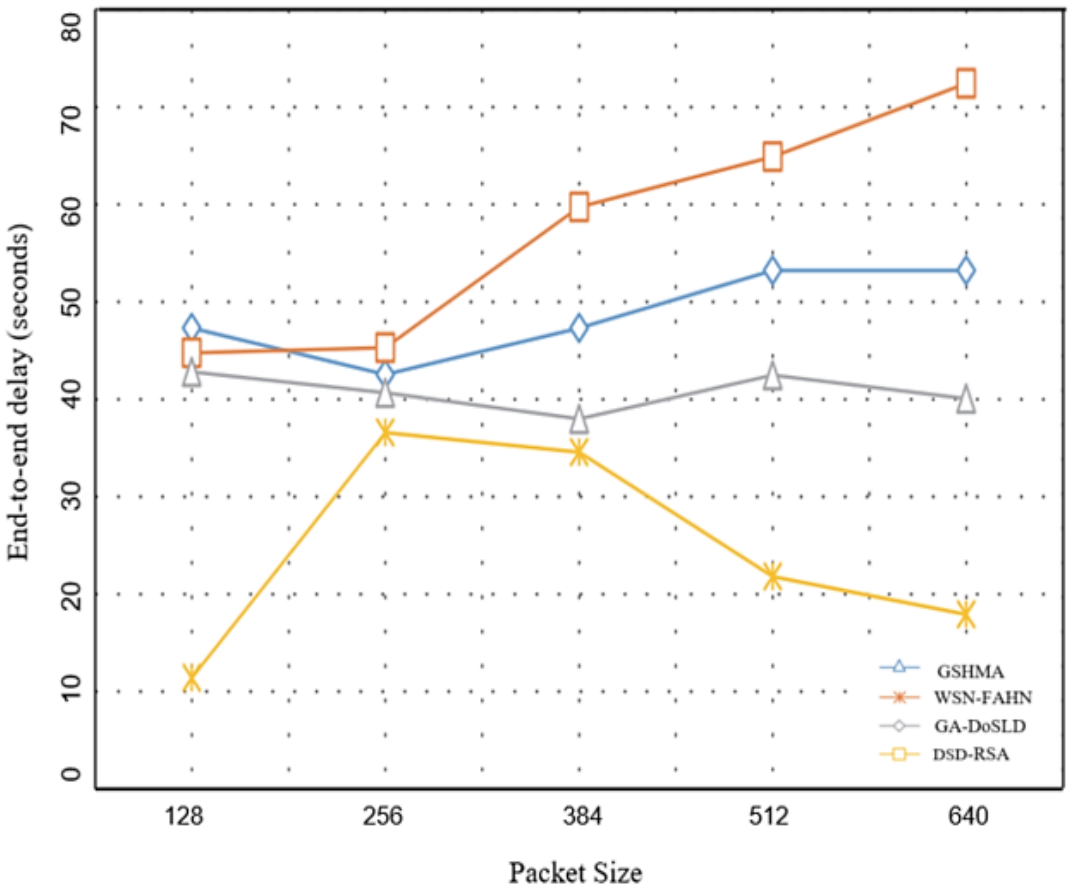


Figure 5.2: Average delay

5.3.3 Energy consumption

The average energy consumption is the amount of energy consumed by the algorithms for transmitting the data packets. The comparison of average energy consumption for the existing GA-DoSL, WSN-FAHN, and GSHMAC algorithms, and the proposed DSD-RSA algorithm is exhibited in Figure 5.3. The graph clearly shows that the DSD-RSA algorithm proposed in the study results in significantly lower energy consumption compared with existing schemes when preventing DoSL. The proposed DSD-RSA, WSN-FAHN, GSHMAC, and GA-DoSLD algorithms experienced an average energy consumption of 30%, 35%, 45% and 50%, respectively.

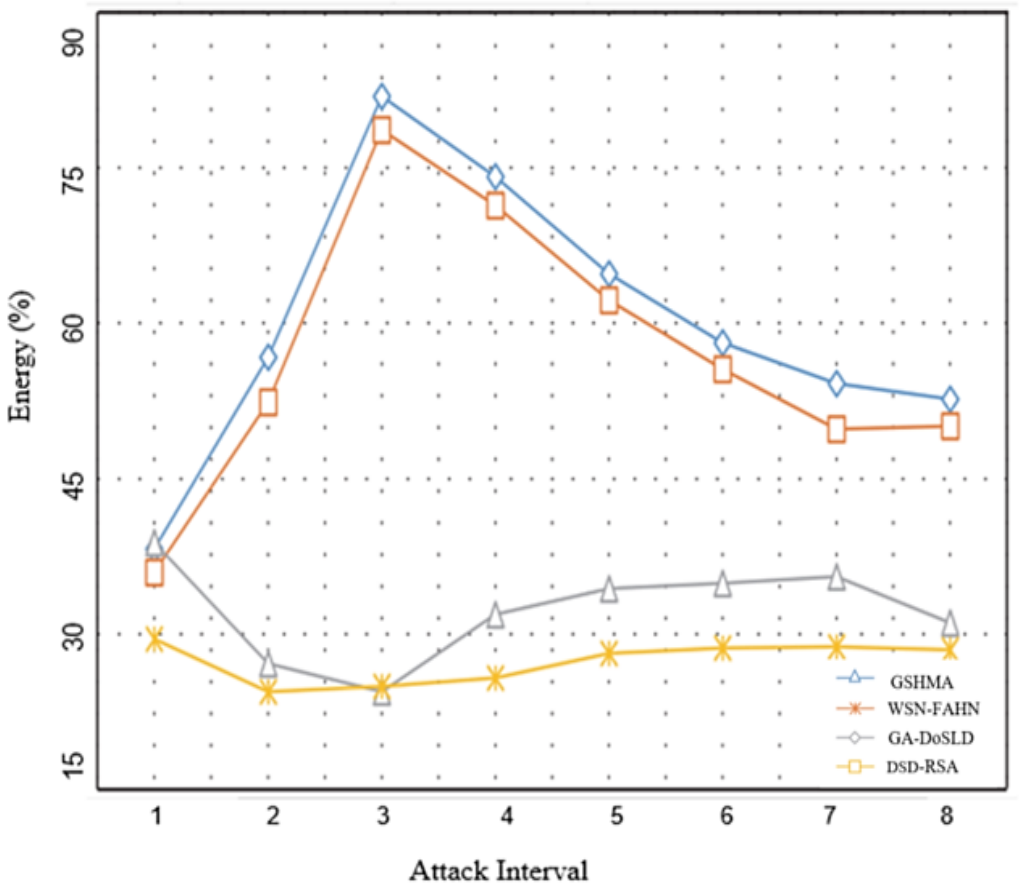


Figure 5.3: Energy consumption

5.4 Chapter Summary

As a result of Section 4.2, this chapter presents and discusses experimental evaluations of the proposed DSD-RSA algorithm. The DSD-RSA algorithm was also discussed in this study as a means of improving WSN performance and QoS. Three performance metrics were analysed and discussed based on the results recorded by NS-2 using trace files. The metrics included average throughput, average end-to-end delay, and energy consumption. Moreover, this study discussed the reason behind this choice, which is that QoS improvements rely on these performance metrics in any network. On the other hand, this study introduced the MATLAB programme and briefly mentioned that it is used to graphically display the results. In the next chapter, the objective achievements, conclusion, and future work of this study are presented wherein the issues not already addressed are briefly discussed.

Chapter 6: Conclusion and Future Work

6.1 Introduction

In this dissertation, we designed and implemented a DSD-RSA algorithm to improve network throughput while reducing end-to-end delay and energy consumption. To achieve the objectives of this study, we designed the proposed DSD-RSA algorithm by integrating the S-MAC protocol, AES, and RSA algorithms to secure WSNs against DoSL attacks, while reducing energy consumption and ensuring QoS in terms of end-to-end delay and throughput.

This chapter is organized as follows: In Section 6.2, we discuss in detail how we achieved the set aim and objectives. In Section 6.3, we provide a summary of this study. In Section 6.4 we discuss the challenges we encountered during the study. In Section 6.5 we then discuss future work. Finally, in Section 6.6 we summarize this chapter.

6.2 Aim and Objectives Achievements

In Section 1.5 of Chapter 1 we discussed how we achieved the set aim and objectives outlined.

1. **To review and identify the existing security schemes currently being used to secure and protect WSNs from DoSL attacks**

In Section 2.10, we identified and analysed existing DoSL algorithms previously used to improve network throughput, and to reduce end-to-end delay and energy consumption during DoSL in WSNs. We discovered that previously proposed DoSL algorithms have failed simultaneously to solve constraints such as throughput, end-to-end delay, and energy consumption. This dissertation proposed the DSD-RSA

algorithm in order to reduce delay and energy consumption while improving the network throughput during DoSL in WSNs. The proposed algorithm was designed by integrating the S-MAC protocol, AES, and RSA algorithms to provide an accurate network that secures WNS over DoSL.

2. **To design and implement an enhanced security model that can detect DoSL attacks in WSNs**

To minimize delay and energy consumption and to improve network throughput during DoSL attacks in WSNs, the proposed DSD-RSA algorithm was designed, as shown in Section 3.3 of Chapter 3. Using Figure 3.1, we presented the typical DoSL WSNs systems architecture. During the design of DSD, we integrated the S-MAC protocol, the AES, and the RSA algorithms to provide an accurate network that secures WNS over DoSL by reducing energy consumption while ensuring QoS in terms of end-to-end delay and throughput. The proposed DSD comprises various types of network components. In Chapter 4, this study presented the implementation of the proposed DSD-RSA algorithm. The implementation was carried out using the NS-2 simulation tool. This study had to modify some C++ files under the NS-2.35 directory. These modifications were specifically meant for message handling. On the other hand, this study made changes to some header TCL files, as well as to the Make file.in file.

3. **To evaluate the effectiveness of the proposed algorithm versus the state- of-the-art-related algorithm in WSNs**

In Chapter 5, this study showed how the proposed DSD-RSA algorithm was evaluated against the GA-DoSL, WSN-FAHN, and GSHMAC algorithms. As mentioned in Section 5.2, the evaluations were based on the three performance metrics considered in this study; these are the average throughput, average delay, and average energy

consumption. This study used the NS-2 to simulate, test, and perform the evaluations of the DSD-RSA, GA-DoSL, WSN-FAHN, and GSHMAC algorithms. The sample simulation topology was illustrated in Figure 4.2. The topology was created using TCL scripts. Those results were automatically recorded in trace files. This study used these results to analyse average throughput, average delay, and energy consumption. This study used AWK scripts to calculate the average of each of these performance metrics. On the other hand, MATLAB was used to graphically display the analysed results obtained during the simulations. Sensor nodes must be powered down and scheduled to go to sleep periodically. However, DoSL attacks prevent them from sleeping. To achieve this, they send useless bogus packets, which turn on their radio (to process the bogus packets). Such depletes their power supply, resulting in an increase in energy consumption that decreases throughput and increases delay in the WSNs. This reduces their lifetime from years to months, or months to weeks, or even from weeks to days. Consequently, the network performance is reduced. To address the identified problem, the study has proposed and developed the DSD-RSA algorithm. When designing the proposed DSD-RSA algorithm, this study has integrated the S-MAC protocol, the AES, and the RSA algorithms to solve the identified problem.

As mentioned in Section 3.2, the proposed network architecture was designed through pooling together various components. This study implemented, tested, and evaluated the proposed DSD-RSA algorithm using NS-2. The NS-2 could be defined as an object-oriented discrete-event network simulator targeted at networking research.

Viewing Section 4.3, NS-2 is seen as the most popularly used simulation tool because it is less complex than other free simulation tools. In Chapter 5, this study presented

and discussed the results and analysis of three performance metrics considered in this work, namely, average throughput, average delay, and energy consumption. This study used the statistical results from the trace files to calculate average throughput, average delay, as well as energy consumption. Also, this study wrote AWK scripts to calculate the average of each of the three metrics.

In this study, MATLAB was used to graphically display the analysed results obtained during the simulations. The study obtained an improved average throughput, average delay, and reduced energy consumption. These promising results yielded a better QoS compared with traditional algorithms.

6.3 Challenges and Future Work

It was very difficult to create the proposed network topology using NS-2. Thus, the researcher had to learn how to use OTcl language and the free source NS-2 topology generator tool. This tool is very efficient, allowing one to drag and drop when defining the network nodes. Moreover, the files generated by the generator tool can be modified at any stage to any desired network topology. Apart from that, it was a very challenging experience to work on two programming languages simultaneously, namely, OTcl and C++. It was also challenging to create a mathematical model in designing the proposed DSD-RSA algorithm. In the future, research on DoSL attacks on WSNs can investigate their integration with other technologies, such as the Internet of Things and cloud computing.

6.4 Conclusion

In this study, the proposed DSD-RSA algorithm was developed and experimentally evaluated using the simulation topology presented in Section 4.3. The evaluation results demonstrated improvements offered by the proposed DSD-RSA algorithm in

optimizing the performance of WSNs in terms of network throughput, delay, and energy consumption. The results were recorded by NS-2 using trace files which were then analysed and discussed apropos of the three performance metrics mainly considered in this study: average throughput, average delay, and energy consumption. The reason behind this choice of metrics is that QoS improvements rely on these performance metrics in any network. Looking at the result discussions and analysis, the proposed DSD-RSA algorithm has yielded promising results. This study has realized an improved average throughput, a reduced average delay, and energy consumption, as demonstrated in Section 5.3. This in return has resulted in an improved QoS being produced by the proposed DSD-RSA algorithm compared with the AES and RSA algorithms. Consequently, the proposed DSD-RSA algorithm has successfully attained the objectives of this study.

References

Abdalkafor, A.S. and Aliesawi, S.A., 2022, October. Data aggregation techniques in wireless sensors networks (WSNs): Taxonomy and an accurate literature survey. In *AIP Conference Proceedings* 24(1), pp. 12-24). AIP Publishing LLC.

Abdulwahab, M.M., Waled, M., Osman, A.A. and Al-Tahir, A., 2022. Wireless sensor networks for real-time monitoring and controlling of poultry farm. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, *14*(1), pp.21-26.

Ahmad, R., Wazirali, R., Bsoul, Q., Abu-Ain, T. and Abu-Ain, W., 2021. Feature-selection and mutual-clustering approaches to improve DoS detection and maintain WSNs' lifetime. *Sensors*, *21*(14), pp.4821-4827.

Al-Ahmadi, S., Aliady, W. and AlRashedy, A., 2022, July. An Efficient Wormhole Attack Detection Method in Wireless Sensor Networks. In *2022 26th International Conference on Circuits, Systems, Communications and Computers (CSCC)*, pp. 240-249.

Almesaeed, R. and Al-Salem, E., 2022. Sybil attack detection scheme based on channel profile and power regulations in wireless sensor networks. *Wireless Networks*, *28*(4), pp.1361-1374.

Appiah, O., Quayson, E. and Opoku, E., 2020. Ultrasonic sensor-based traffic information acquisition system; a cheaper alternative for ITS application in developing countries. *Scientific African*, *9*, pp.00487-004797.

Ayaz, M., Ammad-Uddin, M. and Baig, I., 2017. Wireless sensor's civil applications, prototypes, and future integration possibilities: A review. IEEE Sensors Journal, 18(1), pp.4-30.

Azzabi, T., Farhat, H. and Sahli, N., 2017, January. A survey on wireless sensor networks security issues and military specificities. In 2017 International conference on advanced systems and electric technologies (IC_ASET), IEEE, pp. 66-72.

Bhola, J., Soni, S. and Cheema, G.K., 2019, March. Recent trends for security applications in wireless sensor networks–a technical review. In *2019 6th international conference on computing for sustainable global development (INDIACom)*, pp. 707-712.

Bhushan, B. and Sahoo, G., 2018. Recent advances in attacks, technical challenges, vulnerabilities and their countermeasures in wireless sensor networks. *Wireless Personal Communications*, *98*, pp.2037-2077.

Bi, S., Zeng, Y. and Zhang, R., 2016. Wireless powered communication networks: An overview. IEEE Wireless Communications, 23(2), pp.10-18.

Butun, I., Österberg, P. and Song, H., 2019. Security of the Internet of Things: Vulnerabilities, attacks, and countermeasures. IEEE Communications Surveys & Tutorials, 22(1), pp.616-644.

Campanile, L., Gribaudo, M., Iacono, M., Marulli, F. and Mastroianni, M., 2020. Computer network simulation with ns-3: A systematic literature review. Electronics, 9(2), pp.272-278.

Chan, L., Gomez Chavez, K., Rudolph, H. and Hourani, A., 2020. Hierarchical routing protocols for wireless sensor network: A compressive survey. Wireless Networks, 26(5), pp.3291-3314.

Chowdhury, S.M. and Hossain, A., 2020. Different energy saving schemes in wireless sensor networks: A survey. *Wireless Personal Communications*, *114*(3), pp.2043-2050.

Desai, S., Hadule, P. and Dudhgaonkar, A., 2017. Denial of service attack defense techniques. Int. Res. J. Eng. Technol, 4(10), pp.1532-1535.

Deif, D. and Gadallah, Y., 2017. A comprehensive wireless sensor network reliability metric for critical Internet of Things applications. *EURASIP Journal on Wireless Communications and Networking*, *2017*, pp.1-18.

Dorathy, I. and Chandrasekaran, M., 2018. Simulation tools for mobile ad hoc networks: a survey. Journal of applied research and technology, 16(5), pp.437-445.

Fakhar, F., 2019. Investigate Network Simulation Tools in designing and managing intelligent systems. Journal of Information Systems and Telecommunication, 7(4), pp.278-293.

Fatima, M. and Khursheed, A., 2022. Overview of Resource Management for Wireless Ad Hoc Network. *Smart and Sustainable Approaches for Optimizing Performance of Wireless Networks: Real-time Applications*, pp.91-123.

Farraposo, S., Montenegro, N. and Matias, A., 2014. Evaluation of the role of first-trimester obstetric ultrasound in the detection of major anomalies: a systematic review. Journal of perinatal medicine, 42(2), pp.141-149.

Fotohi, R. and Firoozi Bari, S., 2020. A novel countermeasure technique to protect WSN against denial-of-sleep attacks using firefly and Hopfield neural network (HNN) algorithms. The Journal of Supercomputing, 76(9), pp.6860-6886.

Fouchal, H., Bourdy, E., Wilhelm, G. and Ayaida, M., 2017. A validation tool for cooperative intelligent transport systems. Journal of computational science, 22, pp.283-288.

Gabriel, B. and Onuodu, F.E., 2021. An Improved Model For Wireless Fidelity Using Header Compression Algorithm (HCA). International Journal of Innovative Information Systems & Technology Research, 9(3), pp.131-141.

Gavric, Z. and Simic, D., 2018. Overview of DOS attacks on wireless sensor networks and experimental results for simulation of interference attacks. Ingeniería e Investigación, 38(1), pp.130-138.

Glaropoulos, I., Lagana, M., Fodor, V. and Petrioli, C., 2015. Energy efficient cognitive-MAC for sensor networks under WLAN co-existence. IEEE Transactions on Wireless Communications, 14(7), pp.4075-4089.

Goswami, S., Joardar, S., Das, C.B., Kar, S. and Pal, D.K., 2017. Performance analysis of three routing protocols in manet using the ns-2 and anova test with varying speed of nodes. Ad Hoc Networks, 1, pp.13-17.

He, M., Alba, A.M., Basta, A., Blenk, A. and Kellerer, W., 2019. Flexibility in softwarized networks: Classifications and research challenges. *IEEE Communications Surveys & Tutorials*, *21*(3), pp.2600-2636.

Henderson, T.R., Roy, S., Floyd, S. and Riley, G.F., 2006, October. ns-3 project goals. In Proceeding from the 2006 workshop on ns-2: the IP network simulator pp. 13-18.

Hossain, A., Mondal, S. and Rehena, Z., 2022, December. Determining k-partitions of Mobile WSN ensuring one-hop communication with total coverage. In *2022 IEEE Calcutta Conference (CALCON)*, pp. 196-200.

Ibrahim, D.S. and Zaidan, F.K., 2021. Routing protocols for energy efficiency in WSNs: a review. *SAR Journal*, *4*(1), pp.29-33.

Jan, B., Farman, H., Javed, H., Montrucchio, B., Khan, M. and Ali, S., 2017. Energy efficient hierarchical clustering approaches in wireless sensor networks: A survey. *Wireless Communications and Mobile Computing*, *2017,* pp. 1-13.

Jamali, S. and Fotohi, R., 2017. DAWA: Defending against wormhole attack in MANETs by using fuzzy logic and artificial immune system. the Journal of Supercomputing, 73(12), pp.5173-5196.

Jinhui, X., Yang, T., Feiyue, Y., Leina, P., Juan, X. and Yao, H., 2018. Intrusion detection system for hybrid DoS attacks using energy trust in wireless sensor networks. *Procedia computer science*, *131*, pp.1188-1195.

Goknil, A. and Yildirim, K.S., 2022. Toward Sustainable IoT Applications: Unique Challenges for Programming the Batteryless Edge. *IEEE Software*, *39*(5), pp.92-100.

Khan, W.Z., Aalsalem, M.Y., Gharibi, W. and Arshad, Q., 2016, October. Oil and Gas monitoring using Wireless Sensor Networks: Requirements, issues and challenges. In 2016 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET), pp. 31-35. IEEE.

Khan, T., Singh, K., Abdel-Basset, M., Long, H.V., Singh, S.P. and Manjul, M., 2019. A novel and comprehensive trust estimation clustering based approach for large scale wireless sensor networks. *Ieee Access*, *7*, pp.58221-58240.

Khatkar, M., Kumar, K. and Kumar, B., 2020. An overview of distributed denial of service and internet of things in healthcare devices. *2020 Research, Innovation, Knowledge Management and Technology Application for Business Sustainability (INBUSH)*, pp.44-48.

Kiruthiga, T. and Shanmugasundaram, N., 2021. In-network data aggregation techniques for wireless sensor networks: a survey. In *Computer Networks, Big Data and IoT: Proceedings of ICCBI 2020*, pp. 887-905.

Kocakulak, M. and Butun, I., 2017, January. An overview of Wireless Sensor Networks towards internet of things. In *2017 IEEE 7th annual computing and communication workshop and conference (CCWC)*, pp. 1-6.

Kuang, Y., Ruan, T., Chew, Z.J. and Zhu, M., 2017. Energy harvesting during human walking to power a wireless sensor node. *Sensors and Actuators A: Physical*, *254*, pp.69-77.

Kulkarni, V., Narayana, K. and Sahoo, S.K., 2020. A Survey on Interference Avoiding Methods for Wireless Sensor Networks Working in the 2.4 GHz Frequency Band. *Journal of Engineering Science & Technology Review*, *13*(3), pp. 202-209.

Lata, S., Mehfuz, S. and Urooj, S., 2021. Secure and reliable WSN for Internet of Things: Challenges and enabling technologies. *IEEE Access*, *9*, pp.161103-161128.

Lee, Y., Rathore, S., Park, J.H. and Park, J.H., 2020. A blockchain-based smart home gateway architecture for preventing data forgery. *Human-centric Computing and Information Sciences*, *10*(1), pp.1-14.

Li, Q. and Liu, N., 2020. Monitoring area coverage optimization algorithm based on nodes perceptual mathematical model in wireless sensor networks. *Computer Communications*, *155*, pp.227-234.

Luo, X., 2020. Secure cooperative spectrum sensing strategy based on reputation mechanism for cognitive wireless sensor networks. *IEEE Access*, *8*, pp.131361-131369.

Manishankar, S., Harshitha, S., Mukhopadhyay, A. and Anoop, A., 2019, March. Technologies for network testing: A hybrid approach. In 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC) pp. 946-951. IEEE.

Mehta, D. and Saxena, S., 2020. MCH-EOR: Multi-objective cluster head based energy-aware optimized routing algorithm in wireless sensor networks. *Sustainable Computing: Informatics and Systems*, *28*, pp.100406-100411.

Moler, C. and Little, J., 2020. A history of MATLAB. Proceedings of the ACM on Programming Languages, 4(HOPL), pp.1-67.

Mukherjee, P. and Mukherjee, A., 2019. Advanced processing techniques and secure architecture for sensor networks in ubiquitous healthcare systems. In Sensors for health monitoring, Academic Press, pp. 3-29.

Nguyen, C.T., Saputra, Y.M., Van Huynh, N., Nguyen, N.T., Khoa, T.V., Tuan, B.M., Nguyen, D.N., Hoang, D.T., Vu, T.X., Dutkiewicz, E. and Chatzinotas, S., 2020. Enabling and emerging technologies for social distancing: a comprehensive survey and open problems, pp. 345-352.

Nivedhitha, V., Saminathan, A.G. and Thirumurugan, P., 2020. DMEERP: A dynamic multi-hop energy-efficient routing protocol for WSN. *Microprocessors and Microsystems*, *79*, pp.103291-103297.

Patil, S. and Chaudhari, S., 2016. DoS attack prevention technique in wireless sensor networks. Procedia Computer Science, 79, pp.715-721.

Pawar, P.M., Nielsen, R.H., Prasad, N.R. and Prasad, R., 2018. GSHMAC: green and secure hybrid medium access control for wireless sensor network. Wireless Personal Communications, 100(2), pp.267-281.

Pierson, H.A. and Gashler, M.S., 2017. Deep learning in robotics: a review of recent research. Advanced Robotics, 31(16), pp.821-835.

Premkumar, M. and Sundararajan, T.V.P., 2020. DLDM: Deep learning-based defense mechanism for denial of service attacks in wireless sensor networks. Microprocessors and Microsystems, 79, pp.103278-103284.

Priyanka, E.B., Maheswari, C. and Thangavel, S., 2021. A smart-integrated IoT module for intelligent transportation in oil industry. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, *34*(3), pp. 2731.

Rashid, B. and Rehmani, M.H., 2016. Applications of wireless sensor networks for urban areas: A survey. Journal of network and computer applications, 60, pp.192-219.

Rajaram, M.L., Kougianos, E., Mohanty, S.P. and Choppali, U., 2016. Wireless sensor network simulation frameworks: A tutorial review: MATLAB/Simulink bests the rest. IEEE Consumer Electronics Magazine, 5(2), pp.63-69.

Rajesh, D.M., Srinivas, P.V.V.S. and Lakshmanarao, A., 2022. Intensive Analysis of Routing Algorithm Detection over Mobile Ad hoc Network Using Machine Learning. *IJEER*, *10*(4), pp.963-968.

Raza, U., Kulkarni, P. and Sooriyabandara, M., 2017. Low power wide area networks: An overview. *ieee communications surveys & tutorials*, *19*(2), pp.855-873

Reyna, J.L. and Chester, M.V., 2017. Energy efficiency to reduce residential electricity and natural gas use under climate change. *Nature communications*, *8*(1), p.14916

Rohrer, E., Malliaris, J. and Tschorsch, F., 2019, June. Discharged payment channels: Quantifying the lightning network's resilience to topology-based attacks. In *2019 ieeeeuropean symposium on security and privacy workshops (euros&PW)*, pp. 347-356.

Rostami, A.S., Badkoobe, M., Mohanna, F., Keshavarz, H., Hosseinabadi, A.A.R. and Sangaiah, A.K., 2018. Survey on clustering in heterogeneous and homogeneous wireless sensor networks. *The Journal of Supercomputing*, *74*, pp.277-323.

Saadi, N., Bounceur, A., Euler, R., Lounis, M., Bezoui, M., Kerkar, M. and Pottier, B., 2020. Maximum lifetime target coverage in wireless sensor networks. Wireless Personal Communications, 111(3), pp. 1525-1543.

Sakib, A.N., Drieberg, M., Sarang, S., Aziz, A.A., Hang, N.T.T. and Stojanović, G.M., 2022. Energy-aware QoS MAC protocol based on prioritized data and multi-hop routing for wireless sensor networks. *Sensors*, *22*(7), pp. 2598-2607.

Sadiku, M.N. and Akujuobi, C.M., 2022. Wireless Networks. In Fundamentals of Computer Networks, Springer, pp. 115-135.

Scaife, N., Peeters, C. and Traynor, P., 2018. Fear the reaper: Characterization and fast detection of card skimmers. In *27th {USENIX} Security Symposium ({USENIX} Security, 18,* pp. 1-14.

Sharma, D., Ojha, A. and Bhondekar, A.P., 2019. Heterogeneity consideration in wireless sensor networks routing algorithms: a review. *The journal of supercomputing*, *75*(5), pp.2341-2394.

Sharma, N., Kaushik, I., Agarwal, V.K., Bhushan, B. and Khamparia, A., 2021. Attacks and security measures in wireless sensor network. *Intelligent Data Analytics for Terror Threat Prediction: Architectures, Methodologies, Techniques and Applications*, pp.237-268

Sinde, R., Begum, F., Njau, K. and Kaijage, S., 2020. Refining network lifetime of wireless sensor network using energy-efficient clustering and DRL-based sleep scheduling. *Sensors*, *20*(5), pp.1540-1547.

Singh, S. and Saini, H.S., 2021. Learning-based security technique for selective forwarding attack in clustered WSN. *Wireless Personal Communications*, *118*(1), pp. 789-814

Suryaprabha, E. and Saravana Kumar, N.M., 2020. Enhancement of security using optimized DoS (denial-of-service) detection algorithm for wireless sensor network. *Soft Computing*, *24*(14), pp.10681-10691.

Syafrudin, M., Alfian, G., Fitriyani, N.L. and Rhee, J., 2018. Performance analysis of IoT-based sensor, big data processing, and machine learning model for real-time monitoring system in automotive manufacturing. *Sensors*, *18*(9), pp.2946.

Tian, L., Santi, S., Seferagić, A., Lan, J. and Famaey, J., 2021. Wi-Fi HaLow for the Internet of Things: An up-to-date survey on IEEE 802.11 ah research. *Journal of Network and Computer Applications*, *182*, pp.103036-103042.

Tronci, E.M., Nagakubo, S., Hieda, H., Demo, L. and Feng, M.Q., 2022, April. Low-power multi-hop wireless sensor network for vibration monitoring of large structures. In *Nondestructive Characterization and Monitoring of Advanced Materials, Aerospace, Civil Infrastructure, and Transportation XVI* (Vol. 12047, pp. 376-382.

Vyas, K., Shukla, S., Dsouza, H., D'Souza, D. and Narnaware, V., 2020. A Survey on Environment Monitoring using Sensor Networks. *Asian Journal For Convergence In Technology (AJCT) ISSN-2350-1146*, *6*(3), pp.95-99.

Xenofontos, C., Zografopoulos, I., Konstantinou, C., Jolfaei, A., Khan, M.K. and Choo, K.K.R., 2021. Consumer, commercial, and industrial iot (in) security: Attack taxonomy and case studies. *IEEE Internet of Things Journal*, *9*(1), pp.199-221.

Xu, C., Xiong, Z., Zhao, G. and Yu, S., 2019. An energy-efficient region source routing protocol for lifetime maximization in WSN. *IEEE Access*, *7*, pp.135277-135289.

Yu, Q., Jibin, L. and Jiang, L., 2016. An improved ARIMA-based traffic anomaly detection algorithm for wireless sensor networks. *International Journal of Distributed Sensor Networks*, *12*(1), pp.9653230.

Wang, J., Cao, J., Sherratt, R.S. and Park, J.H., 2018. An improved ant colony optimization-based approach with mobile sink for wireless sensor networks. The Journal of Supercomputing, 74(12), pp.6633-6645.

Zheng, G., Gong, B. and Zhang, Y., 2021. Dynamic network security mechanism based on trust management in wireless sensor networks. *Wireless Communications and Mobile Computing*, *2021*, pp.1-10.

Zheng, Y., Rajasegarar, S. and Leckie, C., 2015, April. Parking availability prediction for sensor-enabled car parks in smart cities. In 2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), IEEE, pp. 1-6.

Appendices

```
N=100; % No. of nodes
t1='400x400'; % Coverage area
Length = 1; %connection radius
Load input.mat
t2=strsplit(t1,'x');
min1=0;
max1=str2double(t2(2));
X = min1+(max1-min1)*rand(1,N);
Y = min1+(max1-min1)*rand(1,N);

%%
figure,
plot(X,Y,'o','LineWidth',1,...
                    'MarkerEdgeColor','k',...
                    'MarkerFaceColor','b',...
                    'MarkerSize',8');
          xlabel('X in m')
          ylabel('Y in m')


for i2 = 1:N

          text(X(i2), Y(i2), num2str(i2),'FontSize',10);
          hold on;
end
hold on
xlabel('x');
ylabel('y');

Rc=400; % coverage Area

% position of sensor node
t1='400x400';
t2=strsplit(t1,'x');
Xb =str2double(t2(1));
Yb =str2double(t2(2));
t1='50x50';
t2=strsplit(t1,'x');
t2=[60 60]; % zone size
gapX=str2double(t2(1));
gapY=str2double(t2(2));
         hold on
         plot(Xb,Yb,'s','LineWidth',1,...
                    'MarkerEdgeColor','k',...
                    'MarkerFaceColor','y',...
                    'MarkerSize',12');
         xlabel('X in m')
         ylabel('Y in m')
         text(Xb, Yb, 'Base','FontSize',10);
         hold on;

%% Ch selection
chalg1='chselalg.m'
chalg=strrep(chalg1,'.m','');
varName=matlab.lang.makeValidName(chalg);
chselalg=str2func(varName)
```

```
[CH]=chselalg(EexL,nodesCH)

%% Path selection %%
alpha=0.01;%0.001 energy per distance %%0.001 %% node to CH power Ratio
beta=0.045;%0.0015        %% CH to sink distance power ratio

Sector1=1;
nodes=N;% Total No. of Nodes
E=10.*ones(1,nodes);
nodesCH=2;% no of cluster head
EexL=E;
E1=E;
if(Sector1==1)
 if(~isempty(pathL))
     path1=pathL;
    for p =1:(size(path1,2))-1
            if(path1(p+1)==-1)
                line([X(path1(p)) Xb], [Y(path1(p)) Yb],
'Color','g','LineWidth',2, 'LineStyle','-')
                arrow([X(path1(p)) Y(path1(p)) ], [Xb Yb ]);
            else
                line([X(path1(p)) X(path1(p+1))], [Y(path1(p)) Y(path1(p+1))],
'Color','g','LineWidth',2, 'LineStyle','-')
                arrow([X(path1(p)) Y(path1(p)) ], [X(path1(p+1)) Y(path1(p+1))])
            end
            hold on
    end

 end
 end
%% Equal-zone division
zX=0:gapX:max1;
zY=0:gapY:max1;
id=zeros(1,N);
ik1=1;
for ik=1:numel(zX)-1
    for ij=1:numel(zY)-1
    rectangle('Position',[zX(ik) zY(ij) gapX gapY]);
         points = [zX(ik) zY(ij); zX(ik)+gapX zY(ij);  zX(ik)+gapX zY(ij)+gapY;
     zX(ik)   zY(ij)+gapY ;zX(ik) zY(ij)];
    plot(points(:,1),points(:,2),'-*r');
    [in,on]=inpolygon(X,Y,points(:,1),points(:,2));
    ind=find(in==1);
    SN(ik1).id=ind;
    ik1=ik1+1;
    hold on
    end
end
xlabel('x');
ylabel('y');


%%%%%%%%%%%%%%%%plotting for Security graph%%%%%%%%%%%%%%%
x= [8 16 32 64 128 256]; %Key Length in bits
p= [97 98 99 98 99 99]; %LSA
y= [96 98 97 96 98 97]; % SPN
r= [94 94 95 94 95 96]; % Feistel
%%%%%%%%%%%%%%%%plotting for LSA%%%%%%%%%%%%%%%%%%%%%%%%
f2=plot(x(1),p(1),'-^');
```

```
axis([ min(x)-1 max(x)+1 min(p)-1 max(p)+1]);
xlabel('Key Size(bits)')
ylabel('Security(%)')
hold on;
for i=2:length(x)
pause(1);
set(f2,'XData',x(1:i),'YData',p(1:i));
end
hold on;
%%%%%%%%%plotting for SPN%%%%%%%%%
fl=plot(x(1),y(1),'-d');
set(fl,'color','red')
axis([ min(x)-1 max(x)+1 min(y)-1 max(y)+1]);
hold on;
for i=2:length(x)
pause(1);
set(fl,'XData',x(1:i),'YData',y(1:i));
end
hold on
%%%%%%%%plotting for Feistel%%%%%%%%%%
fl=plot(x(1),r(1),'-d');
set(fl,'color','green')
axis([ min(x)-1 max(x)+1 min(r)-1 max(r)+1]);
hold on;
for i=2:length(x)
pause(1);
set(fl,'XData',x(1:i),'YData',r(1:i));
end
legend('LSA','SPN','Feistel');
hold off
x=10:100;


%%%%%%%%%%%%%%%%plotting for Key expansion graph%%%%%%%%%%%%%%%
x= [8 16 32 64 128 256];
p= [50 65 70 83 89 102]; %LSA
y= [75 83 89 95 110 140]; % SPN
r= [64 75 87 90 97 115]; % Feistel
%%%%%%%%%%%%%%%%plotting for LSA%%%%%%%%%%%%%%%%%%%%%%%%
f2=plot(x (1),p(1),'-^');
axis ([ min(x)-1 max(x)+1 min(p)-1 max(p)+1]);
xlabel ('key Size(bits)')
ylabel('Key expansion time(mS)')
hold on;
for i=2:length(x)
pause (1);
set (f2,'XData',x(1:i),'YData',p(1:i));
end
hold on;
%%%%%%%%%plotting for SPN%%%%%%%%%
fl= plot(x(1),y(1),'-d');
set (fl,'color','red')
axis ([ min(x)-1 max(x)+1 min(y)-1 max(y)+1]);
hold on;
for i=2:length(x)
pause (1);
set (fl,'XData',x(1:i),'YData',y(1:i));
end
hold on
```

```
%%%%%%%%plotting for Feistel%%%%%%%%%%
fl=plot(x(1),r(1),'-d');
set(fl,'color','green')
axis([ min(x)-1 max(x)+1 min(r)-1 max(r)+1]);
hold on;
for i=2:length(x)
pause(1);
set(fl,'XData',x(1:i),'YData',r(1:i));
end
legend('LSA','SPN','Feistel');
hold off
x=10:100;

%%%%%%%%%%%%%%%%plotting for Energy consumption graph%%%%%%%%%%%%%%%
x=[0 20 40 60 80 100];
p=[386 361 398 434 467 478]; %LSA
y=[706 704 769 809 854 920]; % SPN
r=[705 752 790 842 878 924]; % Feistel
%%%%%%%%%%%%%%%%plotting for LSA%%%%%%%%%%%%%%%%%%%%%%%%
f2=plot(x(1),p(1),'-^');
axis([ min(x)-1 max(x)+1 min(p)-1 max(p)+1]);
xlabel('Number of Nodes')
ylabel('Energy(uJ)')
hold on;
for i=2:length(x)
pause(1);
set(f2,'XData',x(1:i),'YData',p(1:i));
end
hold on;
%%%%%%%%%plotting for SPN%%%%%%%%%
fl=plot(x(1),y(1),'-d');
set(fl,'color','red')
axis([ min(x)-1 max(x)+1 min(y)-1 max(y)+1]);
hold on;
for i=2:length(x)
pause(1);
set(fl,'XData',x(1:i),'YData',y(1:i));
end
hold on
%%%%%%%%plotting for Feistel%%%%%%%%%%
fl=plot(x(1),r(1),'-d');
set(fl,'color','green')
axis([ min(x)-1 max(x)+1 min(r)-1 max(r)+1]);
hold on;
for i=2:length(x)
pause(1);
set(fl,'XData',x(1:i),'YData',r(1:i));
end
legend('LSA','SPN','Feistel');
hold off
x=10:100;

%%%%%%%%%%%%%%%%plotting for PDR graph%%%%%%%%%%%%%%%
x=[0 20 40 60 80 100];
p=[99 98 97 94 93 90]; %LSA
y=[95 93 91 85 81 78]; % SPN
r=[96 93 87 85 79 75]; % Feistel
%%%%%%%%%%%%%%%%plotting for LSA%%%%%%%%%%%%%%%%%%%%%%%%
f2=plot(x(1),p(1),'-^');
```

```
axis([ min(x)-1 max(x)+1 min(p)-1 max(p)+1]);
xlabel('Number of Nodes')
ylabel('Packet Delivery Ration(%)')
hold on;
for i=2:length(x)
pause(1);
set(f2,'XData',x(1:i),'YData',p(1:i));
end
hold on;
%%%%%%%%%plotting for SPN%%%%%%%%%
fl=plot(x(1),y(1),'-d');
set(fl,'color','red')
axis([ min(x)-1 max(x)+1 min(y)-1 max(y)+1]);
hold on;
for i=2:length(x)
pause(1);
set(fl,'XData',x(1:i),'YData',y(1:i));
end
hold on
%%%%%%%%plotting for Feistel%%%%%%%%%%
fl=plot(x(1),r(1),'-d');
set(fl,'color','green')
axis([ min(x)-1 max(x)+1 min(r)-1 max(r)+1]);
hold on;
for i=2:length(x)
pause(1);
set(fl,'XData',x(1:i),'YData',r(1:i));
end
legend('LSA','SPN','Feistel');
hold off
x=10:100
```