A hybrid algorithm to enhance wireless sensor networks security in the Internet of Things.

A hybrid algorithm to enhance Wireless Sensor Networks security in the Internet of Things

By

Ntebatseng Mahlake

(209199335)

Submitted in partial fulfilment of the requirements for the degree

Master of Computing: INFORMATION TECHNOLOGY

in the

Department of Information Technology

at the

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

TSHWANE UNIVERSITY OF TECHNOLOGY

Supervisor: Dr. Mathonsi TE

Co-Supervisors: Dr. du Plessis DP & Dr. Muchenje T

August 2022

Declaration

I Ntebatseng Mahlake, declare that this thesis research study entitled 'A hybrid algorithm to enhance Wireless Sensor Networks security in the Internet of Things', is my work. It is being submitted in partial fulfilment of the requirements for the Master of Computing Degree: Information Technology at the Tshwane University of Technology, Pretoria. I further declare that all sources cited or quoted are indicated and acknowledged using a comprehensive list of references.

Signature Date

…….. ……..

Dedication

I dedicate this study to my family, especially my mother, who gave me support and encouraged me in my moments of weakness to keep pushing until where I am today.

Acknowledgment

First and foremost, I want to thank God for the strength he has given me to carry out this research. Second, I would like to express my thanks to my supervisors, Dr. TE Mathonsi, Dr. DP du Plessis, and Dr. T Muchenje, for their constant direction, warm encouragement, critical comments, patience, and support. Their guidance and suggestions were always kind and constructive. Finally, I would like to express my gratitude to my entire family for their encouragement and motivation.

Publications

Parts of this research study have been submitted and accepted for publication in the following journal and conference proceedings:

Journal:

1. Ntebatseng Mahlake, Topside E. Mathonsi, Deon P. Du Plessis, and Tonderai Muchenje, “A Lightweight Encryption Algorithm to Enhance Wireless Sensor Networks Security on the Internet of Things”. Accepted for publication at the Journal of Communication, August 2022.

Conference proceedings:

1. Ntebatseng Mahlake, Topside E. Mathonsi, Deon P. Du Plessis, and Tonderai Muchenje, “A Hybrid Algorithm to Enhance Wireless Sensor Networks security on the IoT”. Accepted for publication at the 2022 World Congress in Computer Science, Computer Engineering and Applied Computing (CSCE2022), July 25-28, 2022, Las Vegas, USA.

2. Ntebatseng Mahlake, Topside E. Mathonsi, Deon P. Du Plessis, and Tonderai Muchenje, “An Enhanced Hybrid Algorithm to Enhance Wireless Sensor Networks security in the Internet of Things”. Accepted for publication at the 2nd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME’22), November 16-18, 2022, MALDIVES.

Abstract

The Internet of Things (IoT) is an anticipated future technology that promises to connect a massive number of devices over the internet. Wireless Sensor Networks (WSNs) is being regarded as one of the most essential subnetworks of the IoT. Sensor networks are being utilized by IoT to gather, monitor, and send sensitive data across wireless networks. The information transferred through WSNs is easily exposed to cyberattacks hence, data security is critical. In WSNs, the attacker's adverse aims are to hinder and halt the network's effective use, as well as to interrupt network services, rendering them unreachable to the users or providing the users with false feedback. Because the users do not have control over their data transmitted on the wireless medium or stored in the middleware, anyone with internet access can access the stored and transmitted data. This puts data confidentiality, authenticity, and integrity at risk: users with unauthorized access can easily access, alter, and manipulate data in transit. The devices in the IoT architecture have limitations, such as memory constraint, low power, and computational capabilities that affect the enhancement of the WSNs' security in the IoT. Due to their energy consumption, complexity, and the need for several rounds to encrypt, traditional encryption algorithms are computationally costly. This study proposed a Lightweight Security Algorithm (LSA) by assimilating the Security Protocol for Sensor Networks (SPINS) and Secure IoT (SIT) encryption technique in order to improve WSNs' data security; this while minimizing the number of attacks and power consumption in WSNs, without impacting network performance. The proposed LSA was designed in order to protect WSNs by fulfilling security primitives such as data authentication, secrecy, and data integrity. The proposed LSA provided substantial security with moderate computational complexity and power consumption with just five rounds during the simulation process. In addition, the proposed hybrid algorithm prevented data misuse and theft while reducing unauthorized access to the network. Furthermore, the proposed LSA reduced the key generation time by 102mS; thus it improved security by 99%. During data transmission, the power consumption dropped by an average of 411.2uJ, and the Packet Drop Ratio (PDR) was between 90% and 99% when compared with SPN and Feistel techniques.

Keywords: Computationally costly, encryption, IoT, security, WSNs.

Table of Contents

Table of Figures

List of Tables

List of Abbreviations

AES	Advanced Encryption Standard
AMPC	AES and Modified Playfair Cipher
APHA	Aggregated-proof-based Hierarchical Authentication
BSPN	Byte-oriented Substitution-Permutation Network
CBC	Cipher Block Chaining
CH	Cluster Heads
CPU	Central Processing Unit
DES	Discrete Event Simulation
DOS	Denial of Service
DTLS	Datagram Transport Layer Security
ECB	Electronic Code Book
ECC	Elliptical Curve Cryptography
FM	Fix Matrix
G-TBSA	Generalized Triangle-based Security Algorithm
HRM	Hamming Residue Method
HIGHT	High Security and Lightweight
IDEA	International Data Encryption Algorithm
IoT	Internet of Things
IT	Information Technology
IV	Initial Vector
LED	Light Encryption Device

LISP	Lightweight Security Protocol
LS	Left Shifting
LSA	Lightweight Security Algorithm
MATLAB	MATrix LABoratory
MAC	Media Access Control
MD5	Message Digest-5
NS2	Network Simulator 2
NS3	Network Simulator 3
PDL	Protocol Definition Language
PDR	Packet Delivery Ratio
PMIPv6	Proxy Mobile Internet Protocol version 6
RAM	Random Access Memory
RC5	Rivest Cipher 5
RFID	Radio-Frequency Identification
SF	Secure Force
SIT	Secure IoT
SN	Sink Nodes
SNEP	Security Network Encryption Protocol
SPINS	Security Protocol for Sensor Networks
µTESLA	“Micro” version of the Timed, Efficient, Streaming, Loss-tolerant Authentication Protocol
THCA	Two-phase Hybrid Cryptography Algorithm
UDP	User Datagram Protocol

WSNs	Wireless Sensor Networks
WWW	World Wide Web
ULASHWSN	Ultra-lightweight Authentication Scheme for Heterogeneous Wireless Sensor Networks

Chapter 1: Introduction

1.1 Background Information to the Study

The Internet of Things (IoT) paradigm has the potential to be the most exciting and promising field of technology advancement in the future by changing communication perspectives (Burhanuddin et al., 2018). IoT allows any electrical gadget to be connected to the internet. Wireless Sensor Networks (WSNs) are currently being utilized in the IoT for communication between nodes and the internet. However, the internet is open and public, thus there is a high risk of intruder attacks. Hence, most internet communication protocols use high-security authentication techniques for their clustering and routing procedures (Anandhavalli & Bhuvaneswari, 2019). These authentication techniques are, however, resource-intensive for WSNs nodes in the IoT.

The review of literature has shown that the IoT will connect billions of gadgets which will increase the amount of network traffic on the internet. As the amount of data and traffic increases, data security may be affected. Traditional resource-limited environments and constrained devices, such as WSNs nodes and Radio-Frequency Identification (RFID) tags, have weak computing abilities, small memory footprints, and stringent power consumption. In the context of the IoT, small embedded devices with limited computational capabilities are expected to connect to large networks (Pei et al., 2018).

The conventional security algorithm implementation is demanding and exhausting; hence data security is degraded due to inherent delays, the high number of attacks, and the high computational complexity of these algorithms. In addition, most of the WSN deployments in the IoT are deployed in unattended hostile environments for gathering sensitive data or information. Therefore, it is not an option to sacrifice the security and privacy of user data to an unauthorized user with unauthorized access to critical data and information. A certain level of security is essential to protect user data (Burhanuddin et al., 2018).

The implementation of cryptographic protection is needed as the amount of sensitive data and information increases drastically and is easily manipulated and retransmitted. The need for the lightweight cryptographic algorithm has been widely mentioned and discussed

in the existing literature, yet no accurate solution has been proposed and implemented. A lightweight cryptographic algorithm can be implemented in both software and hardware and also in resource-constrained devices to improve the utilization of resources, computational time, power consumption, and security.

Furthermore, due to limitations such as power, memory, computing time, and speed, complete implementation of a cryptographic algorithm is not possible in resource-constrained devices which are normally used in WSNs in the IoT. Thus, WSNs suffer from a variety of security problems and attacks due to their design. It is therefore recommended to use lightweight security techniques to avoid the technical overheads imposed (Aakash & Shanti, 2016).

Many WSNs solutions have been proposed, with the sensor viewed as a feature of the internet linked via nodes. However, in the IoT, sensor nodes are indeed referred to as internet nodes, highlighting the significance of authentication. Data confidentiality and integrity are also crucial in order to maintain data trustworthiness (Usman et al., 2017).

Less computationally complex cryptographic techniques that will deal with data encryption and authenticating messages must be developed in order to enhance the reliability, security, performance, and privacy aspects of wireless devices. Symmetric-key cryptography is the most widely used technique in WSNs. This is because WSNs nodes in IoT are resource-constrained in terms of processing time, resources, computational capabilities, cost, and performance. Thus, in this research study, a lightweight security algorithm (LSA) was proposed to enhance the security of WSNs in the IoT. The proposed algorithm was designed by integrating SIT with SPINS security protocol for better security and power efficiency. SPINS offers Security Network Encryption Protocol (SNEP) and the "micro" version of the Timed, Efficient, Streaming, Loss-tolerant Authentication Protocol (µTESLA) protocols as the basic block components for key distribution in sensor networks. The core SPINS block components, in particular, describe basic primitives for ensuring confidentiality, two-node authentication, data integrity, and weak message freshness. The proposed LSA was designed to address problems such as data security, minimizing the number of attacks, resource utilization, and power consumption. Only fresh and new data is transmitted, no old data is replayed or retransmitted. Therefore, the study took a

proactive approach to reduce the number of attacks by implementing a strong lightweight security encryption algorithm.

1.2 Research Motivation

The IoT is transforming the Information Technology (IT) industry and will be the next technological leap after the internet (Hamza et al., 2020). For communication between nodes and the internet, WSNs are currently used in the IoT. Since the internet is the most popular and open environment, anybody with access can use it. Thus, it is constantly vulnerable to security threats and attacks because users do not have complete control over their data (Pei et al., 2018). As a result, it becomes more difficult to handle the data that is sent and received. Furthermore, the attackers see an opportunity to tamper with or change data in transit and attempt to gain unauthorized access to the nodes. Attackers can also detect interactions between the two nodes. So far, to the best of my knowledge, no credible solution to deal with such attacks has been proposed. Therefore, this research study proposed an LSA that can mitigate and prevent malicious users from connecting with WSNs applications during data transmission.

In addition, the motivation for conducting this research was to develop research abilities by solving research problems in a variety of ways, as well as to contribute to the field of wireless networks. In addition, the researchers wanted to learn more about wireless network applications, lightweight encryption techniques, and the various attack types that exist. Previous research focused separately on data security, power consumption, and network performance in WSNs. Thus, this study looked into all aspects of WSNs.

1.3 Statement of the problem

WSNs are now used by IoT to collect, monitor, and transmit data and confidential information through wireless networks (Pei et al., 2018). Since the users do not have control over their critical data transferred over the internet, unauthorized users with internet access can access such data. The security of data can thus easily be jeopardized or altered as the amount of data transmitted through a wider network grows. Such is especially so in the context of IoT, where small embedded devices such as sensors with poor computational capabilities are expected to connect to a larger network.

Implementing security techniques is time-consuming and selective. Traditional security algorithms degrade data security and privacy due to the higher number of attacks, computational complexity, and higher power consumption (Ahmed et al., 2019). Thus, inherent delays are experienced during algorithm execution, placing the integrity and confidentiality of data at risk. This makes WSNs prone to a variety of attacks and degrades the security of the data that is transmitted over WSNs in the IoT.

1.4 Research Questions

The main research question of the research study is as follows: How can an LSA for WSNs in the IoT be developed by integrating SIT and SPINS?

To have a better understanding of the status of WSNs in the IoT, sub-questions were introduced and are listed below:

1. What are the existing encryption algorithms used in WSNs in the IoT?
2. How can the proposed LSA be designed by integrating SIT and SPINS for WSNs in the IoT?
3. What is the effectiveness of the proposed LSA against existing encryption algorithms?

1.5 Aim and Objectives of the Study

The study's main aim was to develop an LSA for WSNs in the IoT by integrating SIT and SPINS. The following are the study's sub-objectives:

1. To identify and analyse the existing encryption algorithms in WSNs in the IoT.
2. To design the proposed LSA by integrating SIT and SPINS in WSNs in the IoT.
3. To evaluate the effectiveness of the proposed LSA against existing encryption algorithms.

1.6 Research Methodology

To provide answers to the research questions, the experimental methodology was used, applying the following research methods:

1.6.1 Literature review

White papers, conference papers, organizational information websites, journals, books, and other online sources documenting WSNs in the IoT were used to conduct this study.

1.6.2 Modelling and implementation

The proposed LSA was designed using scientific or mathematical models. Thereafter, a code/program was written in order to implement the propose LSA in WSNs in the IoT.

1.6.3 Simulation

The proposed LSA was tested using a network simulator. Many simulation packages are available. A simulation is a powerful tool for analysing and improving networking technologies. The proposed LSA was simulated using a MATLAB avalanche test. MATLAB is a powerful simulation tool that may be used for a variety of tasks including algorithm development, visualization and exploration, modelling, simulation, and prototyping.

Table 1.1: Alignment of Research Questions, Objectives, and Research Methods

Research Question	**Research Objective**	**Research Methods**
What are the existing encryption algorithms used in WSNs in the IoT?	To identify and analyse the existing encryption algorithms in WSNs in the IoT.	Literature Review
How can the proposed LSA be designed by integrating SIT and SPINS for WSNs in the IoT?	To design the proposed LSA algorithm by integrating SIT and SPINS for WSNs in the IoT.	Modelling and Implementation
What is the effectiveness of the proposed LSA algorithm compared with existing encryption algorithms?	To evaluate the effectiveness of proposed LSA algorithm compared with existing encryption algorithms.	Simulation

1.7 Significance and Benefits of the Study

People who are interested in wireless networks and open-source technology platforms will benefit from the research.

The following are some of the areas in which this research will be beneficial:

1. To minimize data theft: the LSA encryption algorithm for WSNs in the IoT has been developed. The algorithm has reduced data misuse and the number of attacks on the wireless network.
2. People will use this study as a learning curve if interested in various lightweight encryption techniques used to protect nodes from attackers.
3. To restore users' confidence in WSNs because their data will be protected from attacks.

1.8 Limitations of the Study

The proposed LSA algorithm was designed to improve data security while reducing the number of attacks, computational complexity, and power consumption, without compromising the performance of WSNs in the IoT. The study did not concentrate on any other factors such as encryption and decryption time, and memory utilization. Instead, the study focused only on the following network parameters, namely, data security, power consumption, and attack reduction. Hardware implementation and performance will be considered shortly.

1.9 Contribution to Knowledge

The contribution of the study:

1. The integration of SIT and SPINS ensured that the proposed LSA was effective and efficient when it comes to data security during data transmission of WSNs in the IoT. This helped to diminish the number of attacks, guaranteeing that only authorized users with permitted access have access to their data. The integration minimized power consumption, and increased data security without negatively impacting network performance. Thus, the proposed LSA ensured that the

cryptographic primitives of confidentiality, authenticity, and integrity have been fulfilled.

2. Using a limited number of rounds on the proposed LSA helped to minimize power consumption and improve data security. The proposed technique does not use the cipher's real key; instead, the LSA uses the S-box to perform non-linear and linear adjustments, resulting in key diffusion and confusion, which enhances the secrecy of the key. As a result, attackers struggled to obtain the key.
3. The proposed LSA prevented attackers from launching a variety of attacks and data theft. The mentioned LSA also prevented unauthorized users from connecting to the network on WSNs in the IoT. The LSA adapted SPINS that helped to authenticate users and verify data transmitted. Because the user has no control over the data transmitted, attacks can be carried out by detecting interaction between two nodes or between sensor nodes and sink nodes, manipulating such without the proper authorization. Therefore, authenticating the users using the MAC reduced the high level of unauthorized access to the data in transit. Moreover, the use of encryption helped to reduce the amount of harm sustained to data integrity and confidentiality.

1.10 Organization of the Dissertation

The remainder of the dissertation is structured as follows: The review of the literature and an overview of IoT, as well as an overview of related works, wireless network protocols, and an overview of WSNs, are presented in Chapter 2. Furthermore, the chapter discusses the various lightweight schemes used to protect WSNs. In Chapter 3, the research study describes the system design and presents the design of the proposed LSA. In Chapter 4, the system implementation of LSA is given. The Chapter also reflects on the experimental setup and the technologies used to implement the proposed LSA. In Chapter 5, the research study presents the evaluated results. Lastly, in Chapter 6, the research study provides a conclusion and suggestions for future work.

1.11 Summary

The background of WSNs was discussed in this chapter. The motivation of this research study was to design an LSA for WSNs in the IoT that is efficient and effective. A set of

research questions has been identified and defined to support the proposed LSA, which the dissertation answers by its objectives. In addition, this chapter presented the research methodology and research methods that the study followed in order to achieve the main aim. A literature review and related works are presented in the next chapter.

Chapter 2: Literature Review

2.1 Introduction

The goal of information security is to provide data protection, integrity, and privacy. Data security is becoming increasingly important as these sensor nodes are vulnerable to attacks in a variety of ways. WSNs are made up of a huge number of sensor nodes that are resource-constrained due to their nature. Thus, WSNs lie open to assault. As a result, the quality and complexity of attacks will continue to rise day by day. The attackers are skilled enough to gain access to the network without the user's knowledge, manipulating or altering data while in transit, thereafter resending the fabricated information to the users. Therefore, the data that has been transferred over WSNs in the IoT must be protected from misuse, threads, and attacks. Security requirements such as data integrity, confidentiality, and authentication are essential to ensure the security of data transmission in modern security techniques (Burhanuddin et al., 2018).

Providing data security in WSNs in the IoT is also especially necessary for those security-sensitive applications that are resource-constrained − this is one of the major concerns of this study. Moreover, the security vulnerabilities, attacks, and threats threaten the security of critical data that has been transmitted over WSNs in the IoT, making users' lives difficult and putting the security of their data at risk.

The remainder of the chapter is organized as follows: Section 2.2 presents a quick overview of the IoT paradigm, the evolution of IoT the architecture of the IoT. Section 2.3 discusses the fundamentals of WSNs in terms of architecture, with the characteristics of WSNs, the difference and the relationship between WSNs and IoT, security requirements of WSNs, constraints of WSNs, and system security threats classification, as well as security vulnerabilities of WSNs.

The types of security attacks in WSNs and their classification are discussed in Section 2.4. Section 2.5 introduces network security principles, and provides a theoretical overview of the security management scheme of WSNs, security protocols, the cryptography approaches of WSNs, and the lightweight cryptography of WSNs, together with their advantages. Section 2.6 provides the existing lightweight cryptographic algorithm of WSNs

with their advantages and disadvantages and their classifications. Section 2.7 provides the related work. The summary of the chapter is presented in Section 2.8.

2.2 The IoT Overview

The IoT facilitates communication and data-sharing across various Smart objects and networks, such as sensors, ZigBee, and Bluetooth. The IoT gives sensory objects computational capabilities, allowing them to interact and communicate with one another over the internet (Porambage, 2018). This section discusses the evolution of IoT, as well as IoT architecture, including security architecture, and IoT security features.

2.2.1 The evolution of IoT

The IoT is a driving force behind the internet's interconnection of millions to billions of devices. The evolution of the internet has been paved by the maturation of the internet and advances in electronic, networking, and communication technology as shown in Figure 2.1 (Porambage, 2018).

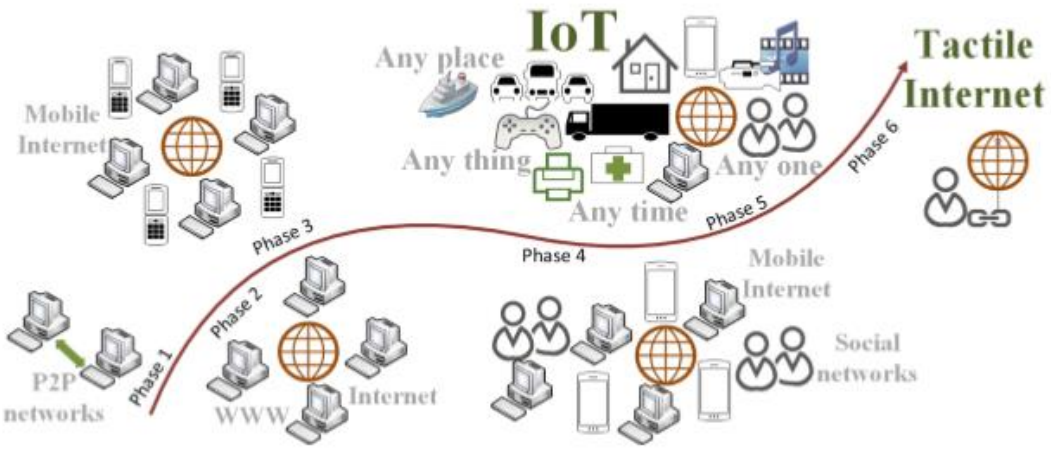

Figure 2.1: Evolution of the internet (Porambage, 2018)

Peer-to-peer connections were used to link two computers in the early stages. The World Wide Web (WWW) was developed in the second process, and it was then used to connect a large number of computers. The mobile internet was implemented in the third process, allowing mobile devices to quickly link to the internet. Then, through the use of social networks, human identities were added to the internet. Since then, the IoT has undergone

a massive paradigm shift as a result of the relation of a diverse set of Smart objects to the internet. Devices and people can communicate with anything and anyone at anytime, anywhere, using any route, network, or service, thanks to the IoT. The tactile internet is believed to be the next significant internet development. Such will provide ultra-responsive, ultra-reliable, and ultra-secure communication links with extremely low latency to IoT networks (Porambage, 2018).

The evolution of the IoT is not only dependent on technological innovation and standardization but also changes in our social perceptions, judgments, knowledge, rules, and laws. For example, in the future IoT age, the way we live as mechanisms or network nodes, and the public display of our activities, could result in a slew of serious security and privacy issues. Standardization, reliability, and durability are also major concerns for IoT growth (Worlu et al., 2019).

2.2.2 Architecture of IoT

The deployed IoT infrastructure is usually heterogeneous. IoT elements, insight, intellect, and flexibility have made it a requirement for future creativity; however, such factors have also rendered the IoT defenceless and unsafe in terms of security. The specific stages of IoT, which are readily accessible, have made it much more difficult for security researchers to find far-reaching answers and solutions to the current security challenges. As a result, the importance of understanding the establishment and the components of IoT becomes critical (Jain & Singh, 2019). The IoT architecture is shown in Figure 2.2, and the layers are explained.

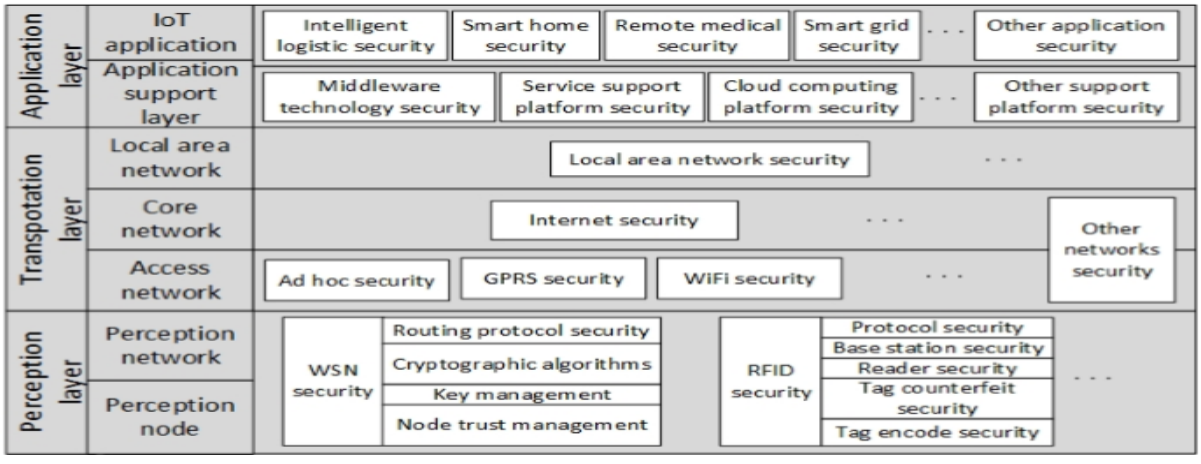

Figure 2.2: Security architecture for IoT (Porambage, 2018)

1. **In the perception layer** of the IoT, cameras, RFID tags, and sensors, among other items, are used to collect data (Porambage, 2018).
2. **Transport layer:** IoT applies the user datagram protocol (UDP) for end-to-end communication. Since the UDP is an insecure protocol, this layer includes a datagram transport layer security (DTLS)-based protection for network devices (Singh et al., 2017). Often known as the IoT nerve centre, the DTLS will aid in the transmission of data gathered by the physical layer. This layer primarily consists of system equipment and items that are primarily focused on administration and data.
3. **Application layer:** This is the topmost layer, in which the real evolution of IoT intelligence is captured. Shopping, social activity, wellness, and personal needs are all possible applications for the application layer (Porambage, 2018).

2.3 Wireless Sensor Networks

The architecture of WSNs, along with its characteristics, the difference between WSNs and IoT, and security, including security requirements to mitigate attacks, and energy consumption in WSNs, are all discussed in this section.

2.3.1 The architecture of WSNs

Ahmed et al. (2019) maintained that WSNs are one of the most important IoT subnets. These distributed networks are made up of small and light wireless sensor nodes that collect various types of data and can be expanded to include physical parameters such as pressure, relative humidity, and temperature. WSNs' sensor nodes have three subsystems: a processing subsystem that computes the sensed information, a sensor subsystem that senses the environment, and a communication subsystem. Each of these three subsystems has its own set of capabilities.

One of the most fundamental technologies used in IoT-based applications is WSNs. The design of WSNs is depicted in Figure 2.3. To monitor and oversee the operation of the desired setting, sensors and actuators are placed inside WSNs. These devices use a wireless medium to connect to a local server for data collection and analysis. Sensor nodes keep an eye on the data they receive, then hop it to other sensor nodes. Several nodes can control monitoring data throughout transmission to the gateway node after multi-hop routing, and subsequently to the management node through the internet or satellite. Authorized

personnel uses the management node to set up and administer WSNs, as well as to collect monitored information and publish monitoring missions (Porambage, 2018).

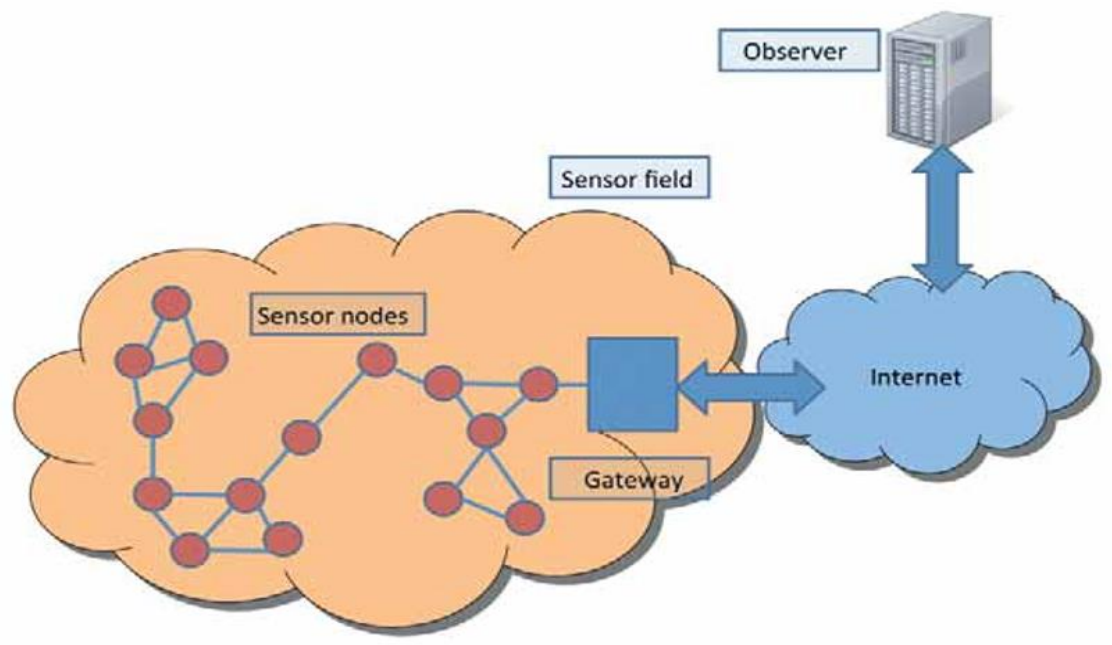

Figure 2.3 Generic WSNs architectures (Porambage, 2018)

In these dispersed networks the sensor nodes have restricted resources, including low processing energy, computational capacities, and low storage capacity. As a result of the limited network resources, a security algorithm is needed that uses less energy and resources while providing high security without sacrificing performance. This kind of security algorithm can also improve data confidentiality and integrity and reduce the chances of unauthorized access (Ahmed et al., 2019). There are numerous characteristics of WSNs for efficient deployment of a sensor network; these should be remembered when it comes to data security. The characteristics are discussed below:

1. Limited resources: because of the nature of the medium of communication and the computing resources of the sensors, which are restricted in terms of power, computational capabilities, and bandwidth, WSNs are susceptible to network attacks (Bangash et al., 2017).
2. Large scale (processing and distributed sending): in WSNs a huge number of sensor nodes are dispersed either randomly or uniformly. All the connected nodes are capable of sorting, aggregating, collecting, processing, and transmitting data to the

sink node. As a result, dispersed sensing ensures system stability and reliability, and the protocols designed for sensor networks should be highly scalable (Sen, 2019).

3. Robust operations: Because WSNs' sensor nodes are normally distributed across a broad network and often in dangerous environments, they operate without human intervention. Such makes it very prone to physical damage, failure, attack, or threats. Therefore, the sensor nodes have to be error and fault-tolerant. Sensor nodes must be able to self-calibrate, self-repair, and self-test (Ahmed et al., 2019).
4. Security and privacy: Each sensor node should have adequate robust security and a mechanism that includes privacy and trustworthiness to prevent unauthorized access, attacks, and unintentional damage of data inside the sensor nodes (Ahmed et al., 2019).

2.3.2 Differences between IoT and WSNs

The approach to this new paradigm shift known as the IoT has been driven by the interconnection of persons and technology. By translating gathered data into Smart data, the IoT implants intelligence and technological expertise into our settings. As a result, WSNs are regarded as a fundamental component of the IoT (Singh et al., 2017). WSNs supply users with the data that the IoT requires. WSNs are usually in charge of monitoring and keeping track of the physical or climate conditions, as well as connecting the collected data to a central location. As a result, WSNs are viewed as a critical component of the IoT. Sensors are found in the first layer of IoT − the perception layer.

2.3.3 Security requirements in WSNs

Ahmed et al. (2019) described WSNs as a unique sort of network because it is sensing and collecting data in different environments using autonomous task-oriented sensors. WSNs are deployed for monitoring the physical world using low-power, low-cost and multipurpose sensors implanted with short-range wireless capabilities. WSNs are vulnerable to security threats due to weak communication and unattended operations. WSNs security services should protect the data and resources transmitted over the network from node attacks and misbehaviour. The research study proposed an encryption algorithm named LSA that ensured data protection of WSNs users' data against security threats and

attacks. The proposed LSA is designed to meet the three security requirements listed below:

1. Data confidentiality: data confidentiality in WSNs prohibits unauthorized parties from viewing data, which is a critical criterion in WSNs operations. Confidentiality refers to restricting access and disclosure of information to authorized users/nodes and prohibiting it from unauthorized users/nodes. Unauthorized people and nodes are unable to access data; whereas approved users and nodes can access data. Restriction of access protects content privacy and renders the information meaningless to any malicious or unauthorized users (Chelli, 2015; Sen, 2019).
2. Data integrity*:* the accuracy, validity, completeness, and trustworthiness of information resources are referred to as integrity. Data integrity refers to data that has not been tampered with either inadvertently, accidentally, or maliciously; data integrity cannot be justified without discovery. In WSNs the sending node is the trusted node from which to receive data, hence source/origin integrity is also included. The full lifespan of data maintenance, consistency, accuracy, and trustworthiness is covered by integrity. Integrity is the preservation of information given by a source, whether incorrect or correct. All information is genuine if it is sent from a reliable source, and if the data received matches the data sent. Even if there are unfortunate events such as power issues, computational inconsistency, or attendance situations, the data security in WSNs against attacks must be prioritized and maintained at a satisfactory level (Bangash et al., 2017).
3. Data authentication*:* because WSNs operate in a public wireless environment, authentication measures are required to detect messages and deceptive packets sent by malicious nodes. Data source and data authentication are key issues: sensor networks are often utilized in unsupervised human environments for essential sensing measurements. WSNs can employ proper authentication procedures to identify sensors and users, safeguarding the integrity and freshness of crucial data, while detecting and preventing impersonation attacks. Authentication mechanisms assist a node in verifying the identity of another node with which it is communicating. If there is no authentication, a malicious node can function as

though it were an independent node, obtaining access to critical data while interfering with the proper operation of other nodes (Sabeel et al., 2016).

2.3.4 Constraints in WSNs

The WSNs are made up of a large number of sensor nodes that are resource constrained by nature. This study proposed a hybrid encryption algorithm named LSA to enhance the security of data transmitted over sensor nodes by reducing the number of threats in WSNs in the IoT. The most common challenges of WSNs are discussed below:

1. Energy constraint: The most significant goal in the design of sensor networks is to reduce power consumption. Reducing sensor nodes power consumption is critical for extending the lifetime of the sensor nodes and network. The sensor nodes are battery-powered devices that are often impossible or difficult to replace or recharge. Additionally, calculations for security functions such as key expansion, signature verification, encryption, and decryption consume a great deal of power. Furthermore, the energy for transmission and management of the security components such as keys, and the storage of the keys, has a high impact on the power or energy consumed by the nodes of sensors (Messai, 2014; Sen, 2019).
2. Denial of Service (DoS) attack: The main aim of this attack is to temporarily disable network resources. Various sorts of DOS attacks can be carried out on WSNs at various layers. Tampering and jamming, for example, are performed at the physical layer; collision and exhaustion are performed at the data link layer; black holes, homing, and misdirection are performed at the network layer; and desynchronization and flooding are performed at the transport layer (Khara, 2017).
3. Unattended operation of the network: The WSNs nodes are typically deployed in remote locations, and thus remain unattended. In addition, WSNs nodes can be deployed in hostile or harsh environments where there is no stable infrastructure. In such an environment, the chances of a sensor being attacked physically are extremely high. Furthermore, continual surveillance after network implementation is challenging. Physical tampering is nearly impossible to detect when WSNs are managed remotely. As a result, WSNs could be vulnerable to a variety of attacks (Mahajan & Sardana, 2017).

4. Security: In WSNs the security of data is forever at risk because of its nature. WSNs are prone to data security attacks and threats, making them untrustworthy to users. The more reliance on information provided by networks, the greater the risk of secure information transfer in wireless networks (Tiwari, 2015).
5. Cryptography: The encryption-decryption algorithms developed for traditional wired networks are sometimes ineffective when applied to wireless networks, particularly WSNs. This is because WSNs comprise small sensors that are severely limited in terms of computation, memory, and battery life. When applying encryption schemes to WSNs, such as the type of keys generated or disseminated, security mechanisms such as encryption may cause an increased number of security threats and delays in WSNs (Tiwari, 2015).

2.4 Security Attacks in WSNs

WSNs are prone to security attacks and threads because of their characteristics and computational limitations (Singh, 2016). This study discusses types of attacks that have a huge impact on violating the security of WSNs applications and data. Security attacks are recognized and classed from several perspectives. These multiple security attacks are discussed below based on their classification:

2.4.1 Attacks on data transit in WSNs

In computing systems, Singh (2016) recognized four classes of system security threats. As data is sent through public networks, it may be vulnerable to a variety of threats, bringing up attack opportunities. Each sensor node wirelessly collects, processes, and delivers data to the database. Because of the scattered and highly sensitive wireless communication, anyone can intercept the network, posing a greater danger to data transmission security. There is a potential for eavesdropping, and data manipulation, among many other attacks. In addition, while users send data, the information in transit may be attacked to provide wrong information to the nodes or sink nodes. There are two types of networks in WSNs, namely, private and public networks. This research study focuses on both public and private networks because anyone with an internet connection and authorized access can access the network. Thus, user data protection is needed. Furthermore, the primary purpose of sensor network security is to protect data, the network itself, the nodes, and

communication between sensor nodes. Attacks that jeopardise and manipulate the data on transit are discussed below:

1. Interruption: In sensor networks, a communication link can be lost or become unavailable, caused by an unknown event. As a result, the service is unable to function properly. The main goal is to carry out DoS attacks. This is directed at the entire WSNs protocol stack (Sabeel et al., 2016).
2. Interception: An interception occurs when an unauthorized entity gains access to the network and its nodes, as well as the data they contain. This puts the secrecy of data transmitted and received at risk. The major goal is to listen to the information communicated. Strong cryptographic techniques have been proposed to solve the above issue. ensuring data security (Sen, 2019).
3. Modification: The data is not only accessed but also tampered with by an unauthorized person in WSNs in the IoT. This jeopardizes the integrity of the message because the users start to question the security of the communication, together with the data transmitted and received. The main goal is to perplex or mislead the communication protocol participants (Sabeel et al., 2016).
4. Fabrication: In WSNs, data security is forever threaded to the users as unauthorized data is inserted, placing the information's credibility at risk. This jeopardizes data integrity and privacy. The main goal is to perplex or mislead the communication protocol participants (Tiwari, 2015).

2.4.2 WSNs' attacks based on attackers' capabilities

In WSNs, an attacker can be classified as shown in Figure 2.4 based on three characteristics: goals, performer, and layer-wise. The section discusses the attacks on WSNs that jeopardize the security of data. These attacks can be broadly categorized as follows:

1. Goal-oriented Attacks

In this section, both passive and active attacks are discussed.

a. Passive attack: These attacks are mostly aimed at compromising data security, especially data confidentiality. Unencrypted traffic is monitored by an attacker in search of sensitive information that can be exploited in other

forms of attack. A passive attack occurs when unauthorized persons monitor and listen in on a communication channel. Monitoring communication decryption, weakly encrypted traffic, obtaining authentication information, and traffic analysis, are all examples of passive attacks. Adversaries can see forthcoming activities by passively intercepting network operations. As a result of such attacks, data or information files are revealed to an attacker without the user's knowledge or consent (Sen, 2019).

b. Active attacks: In active attacks, the attacker is no longer passive but takes proactive steps to gain network control. Active attacks entail tampering with data packets or transmitting non-legitimate packets. The attackers attempt to erase or alter data transferred over the network. The attacker can potentially replay previous messages or data or inject traffic to disrupt network operations (Poongavanam et al., 2018).

2. Performer-oriented Attack

This category of attacks can be either outsider or inside attacks that compromise the security of data in WSNs. Below are details of the attacks:

a. Outsider attacks and inside attacks: Outsider attacks occur when valid inner nodes of a WSNs interact in unauthorized ways; while inside attacks occur when the legitimate inner nodes of WSNs interact in an unauthorized way (Sen, 2019).

3. Layer-oriented Attack

WSNs are layered in structure. Because of their multilayer architecture, these networks are prone to a variety of attacks. Below, various attack categories are discussed:

a. Physical layer attacks: The lowest layer in a WSNs protocol stack is the physical layer. Modulation, encryption, frequency selection, carrier frequency production, and signal detection are all the responsibilities of the physical layer. The primary concern of the physical layer is power efficiency, with

additional issues similar to those of other wireless networks (Bangash et al., 2017).

b. Data link layer attacks: The data link layer is responsible for handling the error corrections, multiplexing of data streams, media access control, and data-frame detections. The data link layer controls the way in which two nodes communicate with each other. The possible attacks on this layer are unjust, malicious collisions, and exhaustion (Sen, 2019).
c. Network layer attacks: WSNs are data-centric networks that save energy by utilizing location awareness and attribute-based addressing. The data link layer controls the way in which two nodes communicate with one another, while the network layer is in charge of determining which node to communicate with. Thus, the network layer is prone to a variety of attacks (Sabeel et al., 2016).
d. Transport layer attacks: When the network system wants to communicate with the outside world, the transport layer comes into action. Because WSNs are not dependent on global addressing, and attribute-based naming is used to define data packet endpoints, communication from the sink to the user is difficult (Sen, 2019).

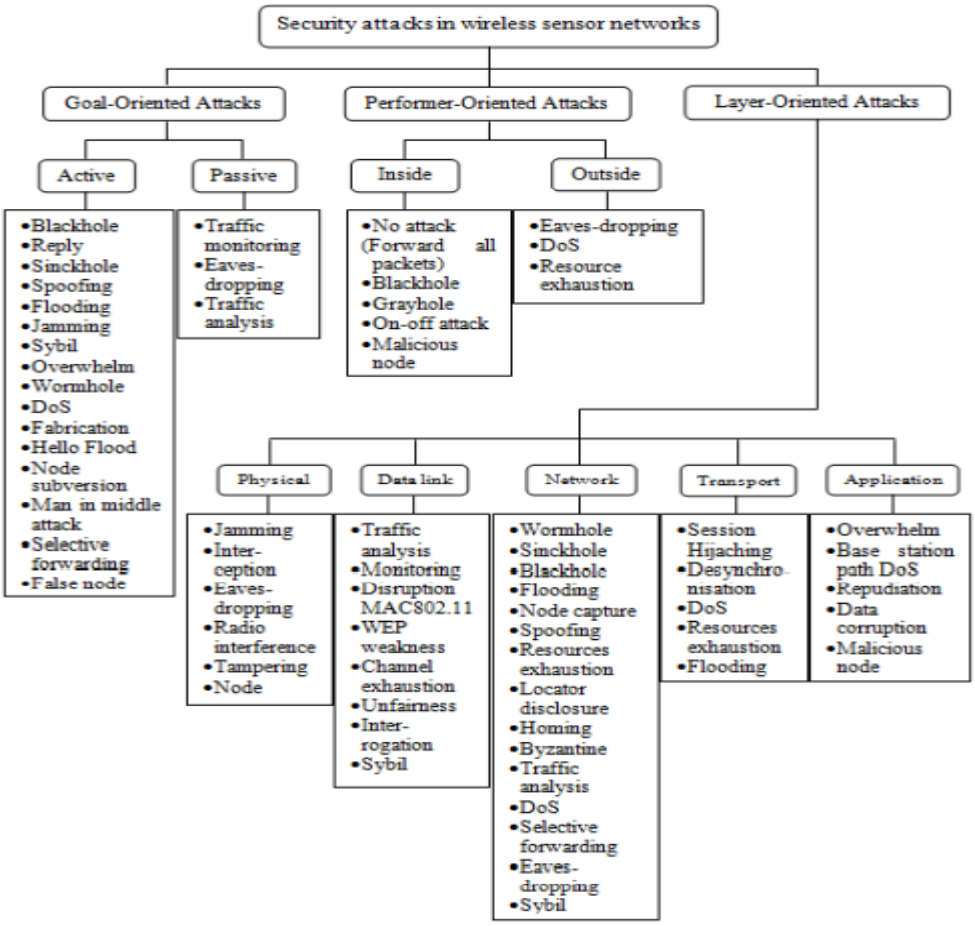

Figure 2.4: Security attacks in WSNs (Chelli, 2015)

2.5 Network Security

Network security is a pressing issue in the communication network. Thus, this section provides an overview of basic network security and related challenges in WSNs in the IoT. Basic network security, cryptography, and its techniques, lightweight cryptography for resource-constrained devices and its benefits, security management schemes, and existing lightweight cryptographic algorithms with benefits and drawbacks, are all explored.

2.5.1 Security management schemes

Security management, often known as network administration, is the procedure for monitoring, controlling, and managing a network's security-related behaviour. Controlling access points to essential or sensitive data held on networked devices is the core job of security management. Security management also entails the smooth integration of several security function modules such as encryption, authentication, and identification, among

many others. Besides these considerations, security management in WSNs should not impose excessive communication, computing, or storage overheads; and it should be consistent with other network management functions (Thakor et al., 2021).

Sabeel et al. (2016) stated that the size of sensors, and hence their memory, processing capability, and kind of responsibilities, represent a fundamental difficulty for adopting any effective cybersecurity strategy in WSNs. This, together with a wide range of threats, makes designing security solutions for WSNs extremely difficult. Security attacks on WSNs can be detected and mitigated using one of the proposed security management schemes. As illustrated in Figure 2.5, these can be divided into two categories: low-level and high-level. However, the study only focuses on the low-level security primitives, discussed below:

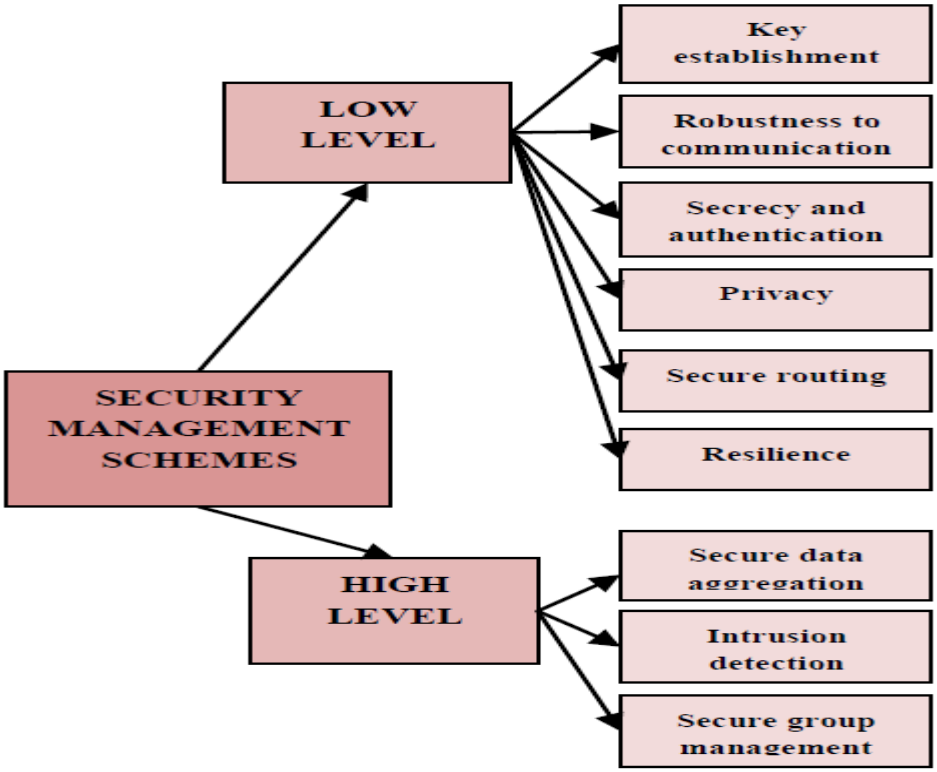

Figure 2.5 WSNs security mechanisms (Sabeel et al., 2016)

1. Low-level mechanism

This section will only focus on the low-level security primitives that will be fulfilled by the proposed LSA that integrates the SIT encryption algorithm with the SPINS security protocol for protecting the data of WSNs in the IoT. The following are examples of low-level security primitives for protecting sensor networks:

a. Key establishment: Due to resource constraints, notably limited power, asymmetric key cryptography should be avoided in sensor networks. The goal should be to set up symmetric key cryptography. In wireless networks with hundreds or thousands of nodes, key-establishment techniques are required. Node keys, cluster keys, and network keys are all examples of keys that can be used. The problem with this technique is that it is not tamper-resistant, which means that attackers can generate all of the keys, thus compromising network privacy (Thakor et al., 2021).
b. Privacy: Sensor networks, as with other traditional networks, must adhere to privacy regulations. There are numerous hazards associated with sensor networks, such as unauthorized individuals gaining access to the network in unexpected ways. It is therefore crucial to provide knowledge of the presence of sensor nodes and data collection (Sen, 2019).
c. Secrecy and Authentication: Sensor network applications, as with other traditional networks, must be protected from eavesdropping, manipulation, attack injection, and packet dropping. The most common method of defence is cryptography. Cryptography improves efficiency while increasing the cost of network setup (Chelli, 2015).
d. Secure routing: The critical services for enabling the sensor networks are data forwarding and routing. Current sensor routing protocols, on the other hand, contain various security flaws, such as network jamming. Node-capture attacks are extremely harmful to sensor networks. In the simplest attacks, malicious routing information is injected into the network, causing routing discrepancies. Simple authentication techniques and safe routing protocols may be developed to protect against such attacks (Sabeel et al., 2016).

e. Resilience to node capture: In WSNs, node capture poses a danger to data security. The majority of applications place sensors in places where attackers can easily access them. An attacker with unauthorized access to the network might seize sensor nodes, obtain cryptographic secrets from them, modify their code, or replace them with malicious nodes under the attacker's control. Tamper-resistant packaging, algorithmic solutions, hashing technique, and obtaining many redundant views of the environment to cross-verify for consistency are some of the protection tactics (Bangash et al., 2017; Chelli, 2015).

2.5.2 Security protocols

WSNs are used in a variety of applications, including monitoring, tracking, and regulating. Security is a critical need for many WSNs applications. WSN security solutions differ from traditional networks due to resource constraints and computational constraints. In this section, WSNs, and security protocols such as TinySec, LSec, Lightweight Security Protocol (LISP), and SPINS are described with their advantages. The SPINS protocol fulfils the security requirement discussed in Section 2.3.3. Thus, this security protocol is used in the proposed solution for better security.

1. TinySec: The TinyOS version includes TinySec, a link layer security architecture developed by the University of Berkeley. Its design is based on ease of use and reduced demand just on the sensor network. Encryption and identity authentication as well as authentication only are the two degrees of security provided by the TinySec. In identity authentication encryption the critical information is encrypted and applies MAC to the package. Packages from the authentication-only are accomplished with MAC since the authentication-only mechanism does not encrypt data. It also meets three requirements: data secrecy, integrity, and authentication (Chelli, 2015).
2. LSec: To provide authentication and authorization, LSec utilizes a basic key exchange method. LSec also includes safeguards to secure data confidentiality, intrusions, and illegal actions. Sensor networks are susceptible to a variety of security risks. Examples of attacks are replay attacks, eavesdropping, DoS, malicious nodes, and message tampering. To combat these types of assaults, LSec utilizes data secrecy, identity verification, data integrity, intrusion defence, and a

variety of security approaches. A comprehensive solution requires strong distribution and key exchange techniques − these difficulties can be addressed partially by encrypting communication between nodes. However, authorization, data privacy, identity authentication, breach of defence, simple secure key exchange, and the use of both symmetrical and asymmetrical encryptions are all provided by LSec (Dener, 2014).

3. LISP: LISP offers security solutions for a wide range of wireless networks with many nodes and few resources. LISP divides large networks into clusters, selects a leader for each cluster, and sets up a key server to expand the clusters. A new switching mechanism has been added to LISP. The mechanism employs a switching technique that entails the use of key servers and head clusters (Dener, 2014). The following are the benefits of LISP:

   It employs an effective key broadcast that does not necessitate the transmission of ACKs;

   a. It uses created bits before incorporating them into the data message.
   b. It employs check bits without including them in the data message.
   c. It has the potential to recover lost keys;
   d. It refreshes keys without encrypting or decrypting data.

   The following are some of the benefits of LISP in terms of protecting critical information against assaults.

   a. Key refreshing protects the network from nodes that can harm it.
   b. Data integrity prevents data from being tampered with during transmissio n.
   c. Access control is accomplished by controlling network inputs.

   LISP is a protocol that is both flexible and energy-conscious. Furthermore, it does not require ACK or other control packages. However, it does not fulfil all the security requirements, such as data freshness.

4. SPINS: SPINS is a set of security protocols designed specifically for sensor networks. Secure Network Encryption Protocol (SNEP) and Micro Timed efficient, Streaming, Loss-Tolerant Authentication (µTESLA) are the two secure building components in

SPINS. While µTESLA authenticates transmissions, SNEP enables data authentication for two parties, data secrecy, and data freshness. While SNEP allows for simple base-to-node communication and vice versa, µTESLA allows for authorized broadcast. Traditionally, asymmetric keys are used to authenticate the initial packets in broadcasts, whereas µTESLA uses symmetric keys to ensure security with symmetric keys disclosure delayed. Unfortunately, synchronization is difficult in a network with multiple nodes (Ndia, 2017). Below are the properties of SNEP that make SPINS the best protocol to integrate with SIT encryption algorithm for protecting data transmitted on sensor networks:

a. Low communication overhead: The communication overhead is reduced by maintaining the counter between the receiver and sender rather than in the message.
b. Semantic security: Because the counter value is incremented after each communication, the same message is encrypted differently every time. The counter value is long enough that it will never repeat during the node's lifetime (Ndia, 2017).
c. Data authentication: A receiver can be confident that the message came from the reported sender if the MAC checks out correctly.
d. Replay protection: The MAC counter value prevents previous communications from being replayed. An opponent might readily repeat messages if the counter was not present in the MAC (Dener, 2014).
e. Weak freshness: The semantic security counter employed between the receiver and sender ensures that the message received is transmitted after the previous one.

SPINS fulfils all the security requirements listed in Section 2.3.3, and it provides a low communication overhead. Thus, it has been chosen to be used in the proposed algorithm.

2.5.3 Cryptography in WSNs

All security functions in WSNs are ensured by cryptography; therefore, selecting the optimal cryptographic solution is crucial. Cryptographic algorithms employed in WSNs should be evaluated based on power consumption, code size, processing time, code size, and data size, as well as resource allocation (Sen, 2019). The cryptography selection process in WSNs is discussed in this section.

2.5.3.1 Symmetric keys cryptography in WSNs vs asymmetric keys cryptography in WSNs

In symmetric cryptography, the same key is used for encryption and decryption; while in asymmetric cryptography, there are two keys, one for encryption and one for decryption, respectively. The main difference is that two unique key pairs are asymmetrical in encrypting and decrypting data.

There are several symmetric and asymmetric cryptographic algorithms designed to provide security in wireless networks. However, due to restricted resource allocation and utilization, most cryptographic algorithms are unreliable for WSNs. Because of their complexity, some studies claim that asymmetric key cryptographic algorithms and some symmetric key cryptographic algorithms, such as High Security and Lightweight (HIGHT) and Rivest Cipher (RC5) are not suitable for WSNs security. Symmetric security algorithms such as International Data Encryption Algorithm (IDEA) and Advanced Encryption Standard (AES) can be used in WSNs in IoT for security. Nevertheless, due to the key size and complexity, these algorithms consume more power. Since the types of nodes used in WSNs are resource-constrained, factors such as power, memory, computational complexity, and execution time, are taken into account (Aakash & Shanthi, 2016).

2.5.4 Lightweight cryptographies for WSNs

Lightweight cryptography can be accomplished on both the software and hardware levels, with chip size, energy consumption at both the hardware and software levels, code size, and Random Access Memory (RAM) complexity in software being used to evaluate the degree of optimization achieved by using lightweight cryptography. Unfortunately, resource-constrained environments such as WSNs need lightweight block ciphers. The required ciphers for the restricted environment do not require high-level security, which is critical on the internet; and security for these applications must be achieved modestly.

Special design techniques, such as memory footprint, execution time, computational capabilities, guaranteed level of security, and resource consumption, must be taken into account to make established algorithms function on resource-constrained devices. The above parameters are used to build and implement lightweight cryptography algorithms from a certain category of cryptography algorithms (Aakash & Santhi, 2016).

The various primitives of lightweight cryptographic algorithms are shown in Figure 2.6. Table 2.1 summarizes a variety of lightweight algorithms based on their key size, block length, number of rounds, and structures (Singh et al., 2017).

The need for lightweight cryptography algorithms has been widely discussed, examining the limitations of the WSNs in the IoT in terms of restricted devices. Indeed, some current lightweight cryptography algorithms do not always take advantage of security-efficiency trade-offs.

According to Usman et al. (2017), in comparing block ciphers, stream ciphers, and hash functions, block ciphers are accepted as significantly more secure and better performing. As a result, when a lightweight cryptographic algorithm is developed and implemented, block ciphers are taken into account. Some criteria must be taken into consideration when developing the block cipher. The block size should be modest to gain the performance benefits of lightweight block ciphers while also saving money. Instead of 128-bits, it should comprise as little as 64-bits. Thus, during the design of the proposed algorithm, the SIT encryption algorithm was selected because it is made up of 64-bits. Such ensures that the security is increasing, while the performance is not degrading.

The size of plaintext is limited when the block size is reduced. In a lightweight block cipher, the key size must be minimal to achieve low power consumption and long battery life. When compared to traditional block cipher algorithms, lightweight block ciphers that target low-resource-limited devices inherently feature simplified computation operations. In lightweight design algorithms, the number of rounds should be limited. A key schedule is a type of algorithm that calculates the sub-keys for rounds for a given key. The implementation of complex keys necessitates additional memory and energy. As a result, a lightweight block cipher uses simpler key schedules when creating sub-keys (Singh et al., 2017).

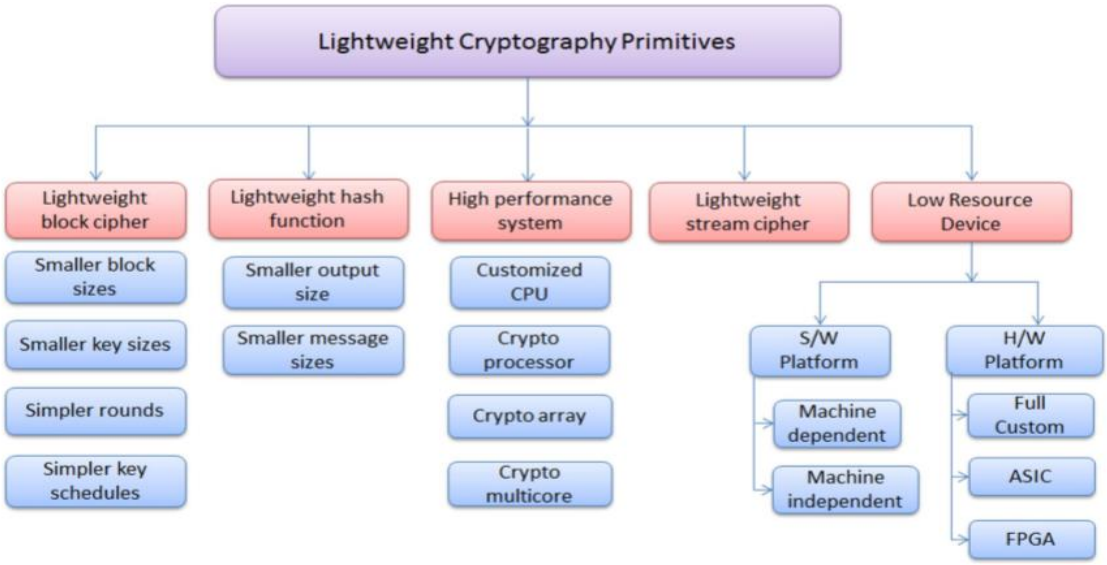

Figure 2.6 Lightweight cryptographic primitives (Singh et al., 2017)

Table 2.1: Summary of some lightweight cryptographic algorithms (Singh et al., 2017)

Algorithm	Structure	Block Size	Number of Rounds	Key size
PRESENT	SPN	64	31	80/128
Pride	SPN	64	20	128
Hummingbird2	SPN	16	4	256
Hummingbird	SPN	16	4	256
Iceberg	SPN	64	16	1900/05/07
RC5	Feistel	32/64/128	1-255	0-2040
AES	SPN	128		128/192/256
TEA	Feistel	64	64	128
XTEA	Feistel	64	64	128
LEA	Feistel	128	24/28/32	128/192/256
HEIGHT	GFS	64	32	128

2.5.5 Significant advantages offered by lightweight cryptography

While implementing cryptography to any resource-constrained IoT devices such as sensor nodes, the three key criteria of lightweight cryptography algorithms to consider are physical cost, performance, and security. The lightweight cryptography algorithms can be implemented on both hardware and software. Table 2.2 lists the three basic characteristics of lightweight cryptography algorithms as well as their benefits. The adoption of one of the six internal structures (SPN, FN, GFN, ARX, NLFSR, and Hybrid) impervious to security threats satisfies the most crucial security characteristic (Thakor et al., 2021).

Table 2.2: The Basic Characteristics and Benefits of Lightweight Cryptography

Significant Advantages		**What advantages may lightweight cryptography provide?**
Cost	Energy consumption	Simple computation with a number of rounds
	Memory(RAM,ROM)	Tiny blocks and key
	Physical area(logic blocks, GE)	Key generation
Performance	Computational power(throughput, latency, execution time)	
Security	Fault-injection and side-channels attacks	Robust internal structure
	Security strength	
	Attack models	

2.6 Existing lightweight cryptographic algorithms with advantages and disadvantages

Various academia, proprietaries, and government entities have presented over fifty symmetric lightweight cryptographic algorithms to lower costs (memory, processing power, computational overhead, physical area (GE), and energy consumption) while improving hardware and software performance (latency, throughput). The structure-based classification of these algorithms is summarized as exhibited in Figure 2.7.

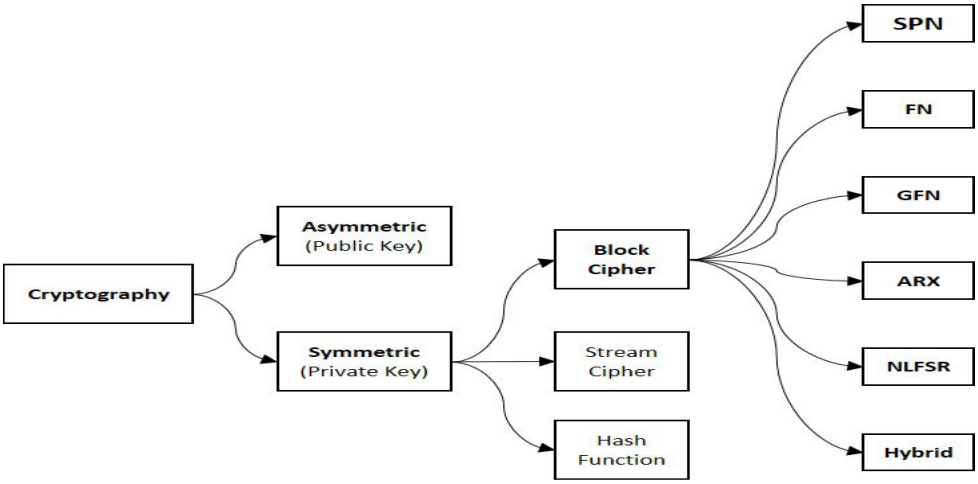

Figure 2.7: Structure classification of lightweight cryptography (Thakor et al., 2021)

In resource-constrained IoT devices, block ciphers are recommended over stream ciphers. According to Sehrawat & Gill (2018), the block and key sizes in block ciphers are both fixed. Confusion and diffusion are two encryption techniques used in block ciphers. The relationship between the encryption key and the cipher text is complex due to the confusion. Every bit of cipher text is expected to be influenced by every bit of key. Diffusion spreads the impact of each bit in a plain text block over a large number of bits in a cipher text block, making encrypted text overly vulnerable to different attacks. Lightweight encryption schemes exist and are currently being used to protect wireless networks. These schemes can be used jointly to maintain a strong and efficient encryption method. This study focuses on block ciphers, specifically symmetric lightweight block ciphers. Existing lightweight cryptographic are methods discussed below:

1. **Substitution-Permutation Network (SPN):** to increase security and expand capacity, lightweight cryptographic schemes are designed and are to be considered as a standard block cipher. SPN are lightweight block ciphers suitable for resource-constraint devices such as sensor nodes, to provide better data security without compromising the performance of the network. For encryption, this method uses mathematical operations. The substitution process is the idea of mixing linear and non-linear operations to observe the relationship between plaintext, cipher text, and the keys. For better data security in WSNs, SPN tinkers with the data, using a collection of substitution boxes and permutation tables, preparing it for the next round. SPN uses a

series of interconnected mathematical operations to ensure that the secrecy of the key and the privacy of the data transmitted is protected from data security threats or attackers.

The number of rounds generated is made up of key mixing, substitution layer, and permutation layer to provide better secrecy of the key with data integrity, data confidentiality, and privacy. A substitution function, also known as a confusion function, creates ambiguity and is part of the substitution/confusion layer in which the attackers lose track of the key. Non-linear operations are supplied by S-boxes or bit-slice implementation in this layer. The permutation layer, also known as the diffusion layer, has a P-box. Invertible linear transformations or simply fixed permutations make up the permutation layer. The permutation layer has additional intrinsic parallelism for confusion and diffusion, and it necessitates the inevitability of the S-box (Sehrawat & Gill, 2018).

Advantages: The operations are commonly performed via XOR/bitwise rotation, with sections of the key being inserted as the rounds progress. The decryption process then reverses the encryption rounds, and the S-boxes/P-boxes operate in the opposite direction.

Disadvantages: Because the SP network has more 'inherent parallelism', encryption and decryption processes can be performed more quickly.

2. **Feistel network (FN):** is the block cipher created to provide confidential communication on the nodes of WSNs. To increase security in WSNs in the IoT, the FN has multiple rounds of encryption. The number of rounds is directly proportionate to the level of security. However, the multiple rounds simultaneously reduce the speed of encryption and decryption. The production of cipher text from plaintext and plaintext from cipher text will be slower as the number of rounds increases.

   FN is the symmetric block cipher that divides the input block into equal halves and applies diffusion to one-half of each half in each round for minimizing power consumption. Additionally, at the start of each round, two halves are

swapped. The Feistel-type structure's decryption function does not have a high implementation cost because it uses the same programme code for both encryption and decryption operations to reduce memory and power needs on the sensor nodes (Sehrawat & Gill, 2018).

Advantages: The fundamental benefit of this approach is that encryption and decryption are virtually identical, if not identical in some circumstances, requiring simply a key schedule reversal. In addition, in FN, the round function does not have to be invertible.

Disadvantages: Feistel structures are easy to integrate into low-average-power hardware (due to the lack of round function in one-half of the states). However, they normally require more round functions for safety reasons; therefore, the higher the number of rounds, the higher the power consumption. In addition, more rounds imply inefficient and time-consuming encryption and decryption processes. Encryption is frequently embedded in applications or utility functions, obviating the need for hardware implementation. As a result, the algorithm's speed of execution becomes an issue; and cryptanalytic vulnerabilities might be raised.

3. **Hybrid:** Hybrid ciphers combine various types of ciphers (SPN, FN, GFN, ARX, NLFSR) or even combine block and stream properties to increase particular features (such as throughput, energy, GE, among many others) dependent on the application needs. SPN and FN are the most popular structures due to their flexibility in implementation based on application requirements. In this study, the SIT algorithm is made up of SPN and informal Feistel. Thus, SIT was integrated with the SPINS security protocol in order to provide better security while lowering the number of attacks in WSNs in the IoT. Although Feistel structures can be easily integrated into low-average-power hardware (owing to the absence of a round functions in half of the states), they often require more round functions than SPN structures for safety reasons (Sehrawat & Gill, 2018; Thakor et al., 2021).

Advantages: Increasing execution encryption and decryption speed while reducing cryptanalytic vulnerabilities.

Disadvantages: Integrating the features of different algorithms sometimes reduces the performance of the network.

2.7 Related Work

Over the past years, WSNs have received much research attention, with most research concentrating on data transmission and reception security. This research study will summarize existing research presented by other researchers, comparing it in order to present research gaps, challenges, and issues that have yet to be addressed. In addition, this research study discusses WSNs lightweight cryptographic approaches that have been previously proposed.

The Hamming Residue Method (HRM) was developed to protect WSNs from malicious attacks. HRM makes sure that the confidentiality of the data is not compromised by unauthorized persons despite the wireless network not being unreliable. Hamming codes are used to discover and repair defects. As a result, various communication systems are familiar with these codes. WSNs are self-contained and consume minimal power; therefore, these codes can be utilized to protect WSNs systems without requiring additional infrastructure. The method was put to the test using the Network Simulator-2 (NS-2) simulation tool. The Hamming residue technique improves the security of WSNs; however, the power consumption is significantly higher (Alotaibi, 2019). Furthermore, the computational complexity was not taken into account. The proposed hybrid technique in this study provided better security while minimizing power consumption and computational complexity.

Shantha et al. (2019) proposed a novel technique named SAT-Jo, which has a new lightweight block cipher. SAT-Jo is an SPN-based technique, suitable for IoT tag-based functionalities. The SAT-Jo technique requires 31 rounds and is based on the SPN block cipher using PRESENT and DES algorithms. By using 24 orders, their proposed technique computes the Galois field's 4x4 S-box; and it utilizes an 80-bit key size with a 64-bit block. This technique provides an appropriate level of security to resource-constrained tag-

based application nodes. However, the network performance was degraded, thus for resource-constrained IoT systems, the proposed technique in this study provided a better balance of resource constraints, security, and performance.

For the WSNs, a lightweight protection algorithm based on Proxy Mobile Internet Protocol version 6 (PMIPv6) was proposed by (Anandhavalli & Bhuvaneswari, 2019). Their proposed security authentication mechanism for WSNs was chosen to replace the inbuilt Diffie-Hellman authentication scheme in the PMIPv6 protocol. A customized Media Access Control (MAC) address-based session key initialization protocol was proposed and tested, together with a seed-based random number session key mechanism. Despite the moderate defence, output was poor, but power consumption was low. The algorithm however, was susceptible to saturation and differential attacks. In addition, computational overhead was not taken into consideration. In the proposed algorithm of this study, the computation complexity was on a moderate level. Thus, the proposed algorithm reduced the number of attacks and power consumption by using the S-Box produced confusion and diffusion to confuse the attackers.

Ebrahim & Chong (2015) proposed a low-complexity cryptographic algorithm for WSNs called Secure Force (SF). The design of the SF algorithm enables a low-complexity architecture for WSNs implementation. Their algorithm is compatible with the proposed algorithm in this study; however, the SF algorithm is not a hybrid. Thus, to save power, the SF encryption procedure consists of five encryption rounds. The SF algorithm is based on the Feistel architecture. The encoding may be performed with a simple design that only employs basic mathematical operations (OR, Shifting, XOR, AND, XNOR, and Swapping) to minimize power usage and enhance data security. This was to generate sufficient dispersion and data dispersal to overcome various types of threats. In addition to data confusion and diffusion, various substitution and permutation techniques are utilized to complicate the cipher. Moreover, the SF algorithm's security, on the other hand, is based on how it works mathematically. The SF algorithm adds to the cipher's complexity by increasing the level of confusion and data diffusion. The SF algorithm increases the difficulty of the cipher by increasing the level of ambiguity and data dispersion. Furthermore, the key expansion technique is highly complex, requiring a wide

variety of mathematical operations that demand more time to execute and consume more power. In this study, using both advantages of the SPN and Feistel assisted in improving the security while protecting the secrecy of the key and reducing the computational time by the left and right shifting.

The proposed Generalized Triangle Based Security Algorithm (G-TBSA) was utilized to combine low-power Wi-Fi into WSNs with the internet to create a novel IoT-based smart home. Their proposed algorithm ensured safe data transmission without compromising the data confidentiality and integrity over a long converge range among many associated sensor nodes in the network. G-TBSA is a resource-friendly data encryption system with a fast key generation method. The algorithm's heart is the key expansion process, which uses fewer resources to generate keys, increasing energy efficiency and reducing complexity. The G-TBSA technique employs a non-right-angle triangle-based technique that relies on the output signal rather than time calculations. The developed IoT-based system performs admirably and meets all of the requisite security requirements. Other algorithms use more energy than the suggested GTBSA (Ahmed et al., 2019). This technique, however, is limited to the sensor's device. Furthermore, the performance factors such as throughput, latency, execution time, and resource allocation were not taken into consideration. In this research, Feistel and uniform SPN architectural solutions are considered for enhancing computational complexity while lowering the number of attacks and boosting data security, particularly data confidentiality and integrity.

The goal of the Ultra-Lightweight Authentication Scheme for Heterogeneous Wireless Sensor Networks (ULASHWSN) project was to achieve secure communication in low-power WSNs in the context of the IoT. The ULASHWSN participates in the user-to-sensor-node authentication process. ULASHWSN defines a straightforward authentication system for multiple communication phases, including registration, key establishment between nodes and users, and authentication between sensor nodes and gateways. ULASHWSN defends wireless sensors from a variety of threats, including replay, impersonation, and DoS. A precise power consumption evaluation was carried out conceptually in ULASHWSN (Khemissa, 2017). ULASHWSN does not disclose network performance metrics like throughput or communication latency.

Chatterjee & Chakraborty (2020) proposed a modified PRESENT in order to design a new lightweight PRESENT cipher that differs from the original PRESENT cipher by altering the key register and shortening the encryption round. The new lightweight PRESENT cipher updates the key register by encrypting its value with the tiny encryption algorithm (TEA) delta value function. The additional layer of the rounds assists in reducing the PRESENT round from 31 to 25, which is the minimal round necessary for protection. Encoding the key register enhances the proposed technique's performance. The suggested approach demonstrates its superiority through the evaluation of several software parameter analyses such as the non-homogeneity, N-gram, histogram, and frequency distribution graph. In terms of gate value, this technique performs better. However, this technique's power consumption was not evaluated.

A lightweight hybrid cryptographic scheme for WSNs called AES and Modified Playfair Cipher (AMPC) was proposed by (Anwar & Maha, 2020) to enhance WSNs security. Their proposed AMPC was designed by utilizing two cryptographic techniques for data security, AES and Modified PlayFair, as well as a third approach, Diffie-Hellman, to secure the key exchange process. Although security was improved, power consumption was quite high. Thus, in this study the proposed algorithm uses a limited number of rounds, thereby reducing power consumption of WSNs in the IoT.

To improve security, Rizk & Alkady (2015) proposed a Two-phase Hybrid Cryptography Algorithm (THCA) for WSNs. To provide encryption, THCA integrated Elliptical Curve Cryptography (ECC) with Advanced Encryption Standard (AES). By integrating asymmetric and asymmetric cryptographic techniques, the THCA security algorithm was intended to provide resilient security with minimum key management. The THCA ensured three cryptographic primitives: integrity, confidentiality, and authenticity. The XOR-DUAL RSA approach was utilized for authentication, and the Message Digest-5 (MD5) technique was utilized for integrity. Their proposed encryption was immune to numerous forms of attacks; however, it consumes much energy. The proposed hybrid algorithm in this study uses a 4x4 S-Box. It has a limited number of rounds; thus it overcomes the aforesaid issue experienced by THCA.

Hamza et al. (2019) proposed a new encryption technique named Chaos-based PRNG to keep patients' information confidential. Their proposed technique generated keyframes immediately after extracting them from video data using a video summarizing technique. The suggested chaos-based technique was used in the symmetric block encryption method, together with a single set of dissemination and confusion processes. Hamza et al. (2019)'s technique employed a unique Pseudo-Random Number Generator (PRNG) based on a two-dimensional logistic map and Zaslavsky's chaotic. Because of its unpredictable effectiveness, their technique was more suitable for real-time applications. The strategy was effective in resisting various efforts to find the secret keys, such as statistical, exhaustion, and differential attacks. The method of these authors was faster and safer than any other currently available algorithm. However, they did not test the power consumption of their technique. Thus, in this study using mathematical operation with the S-Box to create diffusion and confusion with less number of rounds confused the attackers while minimizing computational power.

The challenge with the aforementioned standard data security measures developed for WSNs is that they all use complicated tactics in their respective systems, having too many rounds, and necessitating greater resource usage. As a result, in this study, a security algorithm named LSA was proposed. The proposed LSA was designed based on a simple and efficient data security mechanism. The LSA uses fewer encryption and decryption rounds with different keys for each round to protect the secrecy of the key. The proposed algorithm integrates SPINS and SIT. Using the advantages of both approaches, the proposed LSA provided significant security improvement of WSNs in the IoT. A modest degree of computational complexity was thus maintained, decreased power consumption, reduced the number of attacks, improving data confidentiality and integrity.

Table 2.3: Categories of the Lightweight Cryptography Schemes with Advantages and Disadvantages

Category of the Lightweight Cryptographic Scheme	**Advantages of the Scheme**	**Disadvantages of the Scheme**
HRM Scheme	1. Improved security	1. Power consumption increased

		2. Performance degraded 3. Computational complexity not tested
SAT-Jo Scheme	1. For resource-constrained IoT devices, this technique provides a better combination of security performance and resource constraints.	1. These lightweight algorithms are considered weak, thus vulnerable to key-related and security attacks
PMIPv6 Scheme	1. Moderate defence observed 2. High performance 3. High security	1. High power consumption 2. High resource demand 3. Complexity not tested
SF Scheme	1. Low power consumption 2.Low computational complexity	1. Low performance High resource demand 2. Low security, being prone to linear attacks
G-TBSA Scheme	1. High security 2. Low power consumption and only suitable for WSNs	1. Low performance 2. High computational complexity
ULASHWSN	1. High security; defends wireless sensors from a variety of threats, including replay, impersonation, and denial of service 2. Energy-efficient	1. Low performance 2. High computational complexity

Modified PRESENT Scheme	1. Better performance in terms of gate	1. Energy inefficiency 2. Vulnerable to key-related attacks
AMPC Scheme	1. High security	1. Energy inefficiency 2. Low performance
THCA Scheme	1. High security	1. Energy inefficiency 2. Low performance 3. Computational overhead not tested
Chaos-based PRNG Scheme	1. Resistant to various attacks, such as statistical, exhaustive, and differential attacks for locating secret keys	1. Low performance 2. High resource demand

2.8 Summary

This chapter presented the background information on the architecture of WSNs security in IoT, the overview of WSNs, the characteristics, and security requirements for WSNs, issues, and challenges of WSNs with the security vulnerabilities of WSNs; also the security-management techniques, and the existing WSNs security techniques with their advantages and disadvantages. In addition, we provided the WSNs security algorithms according to their methodology used, discussing several identified gaps from the existing WSNs security algorithms. Moreover, we provided reasons for the LSA algorithm being proposed.

Chapter 3: System Modelling

3.1 Introduction

There is an increasing number of mobile users demanding improved data security and the best network connectivity at any given time, anywhere. This has led to the evolution of WSNs in the context of IoT to fulfil the users' demands. IoT is now using sensor networks to acquire, analyse, and transmit information and sensitive data across wireless networks (Bangash et al., 2017). The security of data can be compromised as the amount of data transmitted through a wider network grows. The resource-constrained environment and constrained devices, such as wireless sensor nodes, have poor computational abilities, small memory footprints, and strict power consumption.

This section illustrates how data is gathered and transmitted from the user to the sensor nodes, sensor nodes to sensor nodes, and sensor nodes to sink nodes. Researchers in this study were able to build an effective encryption algorithm that prevents data misuse and theft in WSNs in the IoT. In addition, the proposed security algorithm named LSA enhances data security and reduces power consumption in WSNs in the IoT without compromising network performance. The LSA is divided into several phases to ensure that data is only accessible by authorized users and comes from the trusted list. The technique is divided into three phases: key expansion, key management, and encryption phase.

In this chapter, the design of the proposed LSA, together with the related presumption, is presented. The LSA integrates the SIT encryption algorithm with SPINS security protocol to enhance security and lower number of the possible attacks. In Section 3.2, the system architecture of a typical WSNs infrastructure is provided. In Section 3.3, the LSA design, with the key expansion process from SIT algorithm, the key management process using SPINS security protocol, the encryption process, the flowchart, and the pseudocode for the LSA algorithm, are provided. Finally, in Section 3.4, a chapter summary is presented.

3.2 System Architecture

The WSNs have become a reality due to recent advances in wireless communications and electronics, which have driven the development of low-cost, low-power, multi-functional

sensors that are compact and can communicate over short distances. Microcontrollers, memory, and a transceiver are all found in each node. The microcontrollers in the sensor node are used to perform tasks, process data, and assist the performance of other components. Memory is primarily used for data storage, whereas the transceiver performs both transmitter and receiver tasks (Othman & Shazali, 2019).

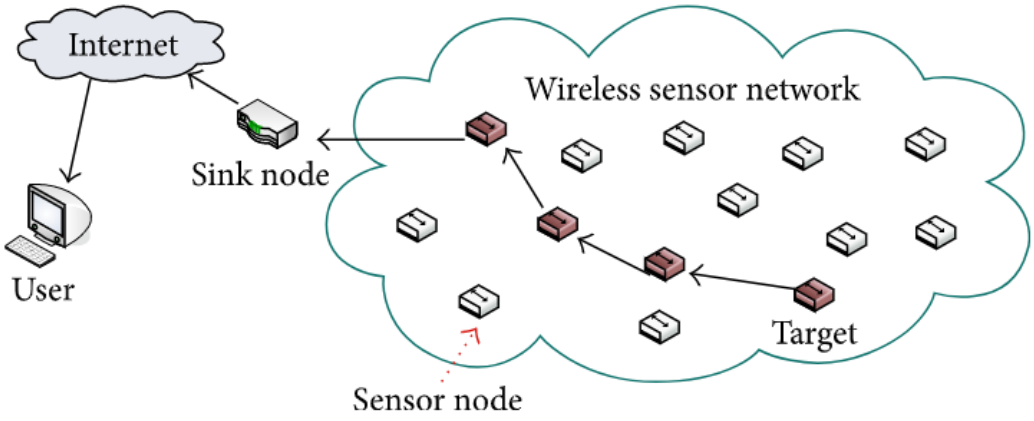

Figure 3.1: Typical WSNs system architecture

Figure 3.1 shows the WSNs system architecture environment that is used for monitoring. Thus, the WSNs system architecture is made up of network devices such as sensor nodes, user nodes, and sink nodes. Sensor nodes communicate with one another and send processed data to a sink node over a wireless connection. The sink node receives data from all nodes, analysing it before sending it to the user over the internet. The attacker can easily create a malicious node to send fake messages to the receiver. The target node sends the packets to the other nodes without the knowledge of unauthorized users. The target node can manipulate the data before it reaches the destination because wireless communication is not secure. The design of the proposed LSA algorithm is presented in the following section, which includes a system model and pseudocode.

3.3 Lightweight Security Algorithm Design

When designing the proposed LSA technique, the research study incorporated existing methods, notably SIT and SPINS, to minimize power consumption and enhance data security during data transmission while minimizing the number of possible attacks on

WSNs in the IoT. Furthermore, the technique prohibits rogue users/nodes from connecting to the WSNs while data is being transmitted.

The key expansion process is the first stage in the proposed approach for WSNs. The keys utilized in the decryption/encryption process are generated during the initial phases. Following the key expansion procedure, the SPINS security protocol performs key management checks, such as determining whether or not the key was disclosed to the attackers. Authentication, confidentiality, and integrity are all verified and reviewed. If the key has been compromised, it is abandoned, and a new key is generated via the key expansion block.

If the key is not accessible to an unauthorized user or rogue node, the key management technique ensures that the network has enough power to carry data from sensor nodes. If there is inadequate power on the network, packets will be dropped, making the network weak and vulnerable to various attacks. If adequate power is available, the encryption process begins, during which the data is secured from attacks, and the computational complexity and protection of information are kept to a reasonable level.

3.3.1 Key expansion

The primary method of generating various keys for encryption and decryption is key expansion. To produce confusion and diffusion, many procedures are carried out. Such actions lessen the likelihood of a weak key while also increasing key strength. The key schedule is used to derive the round keys (K_r) from the input cipher key. The procedure is divided into two parts: key expansion and round key selection. Logic operations (XOR, XNOR), Left Shifting (LS), matrix multiplication using Fix Matrix (FM), permutation using P-table, and transposition using Q-table are all performed by the key expansion. The key is the most crucial part of any encryption or decryption method. Data security is jeopardized if this key is discovered by an attacker. The entire security of the data is dependent on this key. To prevent the potential of a weak key and to improve key strength, various activities are carried out to induce confusion and diffusion.

To preserve security against exhaustive search attacks, the genuine key k_t must be sufficient to limit the enemy's ability to conduct 2^{K_t-1} encryptions for key-searching attacks. The SIT encryption technique is a 64-bit block cipher, which implies that it encrypts 64-bits of data with a 64-bit key. A 64-bit encryption key (K_c) is taken as input from the user. This key K_C will be used as the key expansion block's input. The block will generate five distinct keys after performing considerable operations to cause confusion and diffusion in the input key. These keys will be used in the encryption and decryption processes, and they must be strong enough to stay undecipherable during the procedure.

According to Usman et al. (2017), the block implements an *f*-function that is influenced by the Khazad block cipher that has been altered. Khazad is a wide-trial cipher that is not a Feistel cipher. The wide trial approach is made up of multiple linear and non-linear transformations that assure a complex relationship between output bits and input bits. The following is a detailed explanation of the components of the key expansion process:

1. The 64-bit cipher key (K_C) is split into segments of 4-bits.

2. The *f*-function works with data that is 16-bits long. As a result, four *f*-function blocks are utilised. After executing an initial substitution of segments of cipher key (K_C) as described in Equation (3.1), these 16-bits for each *f*-function are acquired.

$$K_{bif} = ||_{j=1}^{4} K_{c4(j-1)+i} \quad (3.1)$$

where i = 1 to 4 for the first 4 round keys.

3. The next step is to obtain K_{aif} by using the *f*-function to send the 16-bits of K_{bif}, as stated in Equation (3.2).

$$K_{aif} = f(K_{bif}) \quad (3.2)$$

4. P and Q tables make up the *f*-function. These are the tables used to perform linear and non-linear modifications to generate confusion and diffusion as shown in Figure 3.2.

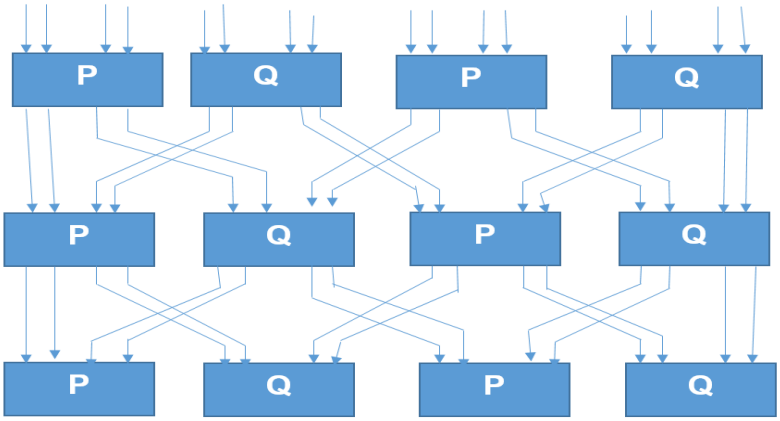

Figure 3.2: f -Function for SIT algorithm (Usman et al., 2017)

Tables 3.1 and 3.2 illustrate the transformations performed by P and Q.

Table 3.1: P Table

Kci	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
P(Kci)	3	F	E	0	5	4	B	C	D	A	9	6	7	8	2	1

Table 3.2: Q Table

Kci	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Q(Kci)	9	E	5	6	A	2	3	C	F	0	4	D	7	B	1	8

5. The matrices K_1, K_2, K_3, and K_4 are translated into four 16-bit arrays to provide round keys.

6. To obtain the fifth key, an XOR operation is conducted among the four round keys, as indicated in Equation (3.3).

$$K_5 = K_1 \oplus K_2 \oplus K_3 \oplus K_5 \quad (3.3)$$

3.3.2 Key Management

SPINS secures two types of sensor network communications, communication from a sink node to any sensor node, and communication from a sensor node to a sensor node. SPINS

in WSNs specifies basic primitives for ensuring confidentiality, authentication between two nodes, data integrity, confidentiality, and weak message freshness. SPINS in WSNs do not ensure broadcast security (Jan et al., 2017). It uses MAC for authentication and integrity, the counter mode for secrecy and confidentiality, and a counter for data freshness. SPINS in WSNs use extremely little memory, low power, and a very low computational overhead. In the proposed algorithm, the SPINS was implemented to check and perform security checks on the key generated from the key expansion block (K_1,K_2, K_3, K_4, and K_5) used in the encryption process taking place between the nodes or between nodes and sink. SPINS makes sure that all the security requirements mentioned in Section 2.3.3 are fulfilled. Electronic Code Book (ECB) attacks are avoided by using a counter value.

The goal of SPINS using counter mode rather than Cipher Block Chaining (CBC) mode is to prevent ECB attacks, which require a random or secret Initial Vector (IV) in CBC mode. However, sending distinct IVs for each transmission adds overhead to the data packet, increasing the sensor node's power consumption. Because the counter value in the CTR mode does not have to be random, it can be known and communicated in advance by the sender and recipient. Moreover, another incentive to use the CTR mode is that the cipher text is the same size as the plaintext, rather than being a multiple of the block size (Jan et al., 2017).

When a CBC Message Authentication Code (CBC-MAC) is used to generate a MAC, IVs are not necessary to achieve authenticity and integrity. Thus, there is no problem of having to send huge IVs wirelessly. The sender and recipient exchange a counter value, which is also used to generate the MAC. Using a counter here protects against old messages being replayed. The message *M* that will be transmitted from *S* (which is the sender) to *R* (which is the receiver) when using SPINS looks as presented in Equation (3.4):

$$S \rightarrow R: \left\{E_{(K\_enc,C)}(M), MAC_{K\_mac}\left(C \middle| E_{(K\_enc,C}(M)\right)\right\} \quad (3.4)$$

where $E_{(K\_enc,C)}(M)$, means the encryption *E* with encryption E_{K_enc} using the counter value *C* on message data *M*. $\left(C \middle| E_{(K\_enc,C}(M)\right)$, concatenate counter value C with $E_{(K\_enc,C}(M)$. $MAC_{K\_mac}\left(C \middle| E_{(K\_enc,C}(M)\right)$ is the authentication code function message MAC key K_mac with (CBC-MAC) on$E_{(K_enc,C}(M)$.

Plain SPINS merely ensures a limited level of data freshness by enforcing a sending order on messages within node *R.* However, it provides no guarantee to node *S* that a message was sent by *R* in response to an event in node *S*.

Node *S* provides high data freshness for a response from node *R* by using a nonce N_S (a random number that is too lengthy to search exhaustively for all conceivable nonces). Node *S* creates N_S at random and transmits it to node *R* along with the request message R_S. In an authenticated protocol, the simplest approach for achieving strong freshness is for *R* to return the nonce with the response message R_R. Instead of returning the nonce to the sender, the proposed algorithm managed to speed up the procedure by incorporating the nonce within the MAC computation.

$$S \rightarrow R: N_{S,}\, R_S$$

$$R \rightarrow S: \{R_R\}\langle K_{RS,} C_R\rangle, MAC(K_{RS}, N_S||C_R||\{R_R\}_{\langle K_{RS} C_R\rangle} \quad (3.5)$$

Node *S* knows that node *R* exists if the MAC is accurate. After sending the request, it generated the response. If the user chooses, the user can use plain SPINS (as specified in Equation (3.4)) for the initial message. Data authentication and secrecy are required; Equation (3.4) will fulfil equation (3.5) requirements. The SPINS algorithm ensures that there is enough power on the network before it transmits data. If the $E_{network}$or E_{round} is insufficient, the packets are discarded − the weaker the network the more chance for attackers to manipulate the data. If the network node is considered alive when the energy node is either greater or equal to or higher than the minimum energy threshold, the process of data transmission continues and ensures that the packets are safe from attack (see Equation (3.6)).

$$E_n \geq E_{th}, \forall_n \in N \quad (3.6)$$

Transmission from the nodes to the Cluster Heads (CH) or Sink Nodes (SN) and data receipt at the CH or SN are required for the total energy. The amount of energy required for transmission from the SN to the nodes is specified. The total energy consumption for each round E_{round} in the network is the sum of the energy consumed in each cluster. Equation (3.7) shows how the energy per round is computed.

$$E_{round} = \sum_{\forall_c \in C} E_{Cluster}$$

$$E_{round} = \sum_{\forall_c \in C}\left[\sum_{\forall_n \in C}\left(E_{T_C - CH} + E_R\right) + E_{T_{SN}}\right] \quad (3.7)$$

The network's total energy utilization $E_{network}$ is computed by deducting each node's remaining energy from the network's total energy; this is calculated as presented in Equation (3.8):

$$E_{network} = \frac{N.E_0 - \sum_{\forall n \in N} E_n}{N.\ E_0} * 100 \quad (3.8)$$

3.3.4 Encryption process

The encryption process begins after the round keys (K_1,K_2, K_3, K_4, and K_5) have been generated. This technique includes some logical operations such as left shifting, swapping, and substitution to create confusion and diffusion. To produce confusion and diffusion, simple processes like XOR, XNOR, AND, OR left shift (LS), swapping, and substitution (S-boxes) are used throughout the encryption process. For the attackers, these processes add complexity and confusion. Below, we present the encryption process steps:

1. The 64-bit block input is divided into four 16-bit sub-blocks, P1, P2, P3, and P4 (PX*0–15*, PX*16–31*, PX*32–47*, and PX*48–63*). This is done to create segments (R1_1, R1_2, R1_3, R1_4).
2. As the bits proceed through each cycle, a swapping operation reduces data originality by changing the order of bits, effectively creating cipher text confusion. Working key sub-keys (K_1, K_2, K_3, K_4 and K_5) are used to address each sub-block by combining operations from several algebraic groups, such as AND, OR, XOR, and XNOR operations to confuse the attackers.
3. The product R1_1 feeds f-Function to produce EFL-1. R1_1 is the output of XNOR between PX 0–13 and K1.
4. The product R1_4 feeds f-Function to produce EFR-1. R1-4 is the output of XNOR between PX 48–63 and K1.
5. The f-Function contains the activity of substitute (S-boxes), AND, OR, and left shift (LS), in addition to the f-Function shown in Equation (3.1). This activity helps in minimizing computational complexity and computational execution time.

$$F = F1 + F2 \rightarrow 32\ Bits$$

$$F1 = OR\left(S - Box\left(AND\left(LS\left(\frac{16\ Bits}{4}\right)\right)\right)\right) \rightarrow 16\ Bits$$

$$F2 = OR\left(S - Box\left(AND\left(LS\left(\frac{16\ Bits}{4}\right)\right)\right)\right) \rightarrow 16\ Bits$$

6. The f-function result is then XORed.
7. R1_2 is the result of XOR between PX 16–31 and EFL-1.
8. R1_3 is the result of XOR between PX 32–47 and EFR-1.
9. During the encryption process, a switching procedure occurs between the two internal parts. The switches are then located between (R1_1 and R1_2) and (R1_3 and R1_4).
10. All the previous processes are designed to increase the complexity of the encryption and to improve the execution time while minimizing computational energy. Equation (3.9) repeats the identical steps for all of the rounds.

$$P_{x\,i,j} \odot K_i; j = 1...4$$

$$R_{i,j} = P_{x\,i,j+1} \oplus Ef_{li}\ ; j = 2 \qquad (3.9)$$

$$P_{x\,i,j-1} \oplus Ef_{ri}\ ; j = 3$$

11. The encoded text is then obtained using Equation (3.10).

$$Ct = R_{5\_1} || R_{5\_2} || R_{5\_3} || R_{5\_4} \qquad (3.10)$$

Some of the notations and mathematical operations used are explained below:

1. Transformation rounds: Every round of the LSA algorithm also contains a variety of transformation operations involving XOR, XNOR, *f*-functions, and swaps.
2. Swapping function: This is the encryption part, in which the left half of 32-bits is switched to the right position, and the right half of 32-bits is swapped to the half-

place left. The basic purpose of changing roles is to change the data's beginning location, to create a more intricate cipher.

3. F-Function: This is a crucial part of the encryption technique that results in data dissemination. Processes of AND & OR, as well as left shifting (LS) operations of 16 data bits, are carried out in the S Boxes.
4. Left Shifting: The LS process divides 16-bits of data into four 4-bit blocks, with left shifting performed on each block. Full data Bit Mixing is one of the activity's outcomes.
5. Substitution(S-Box): The *f*-function utilizes four different types of S-boxes, which are created by combining data and adding complexity to the outcome via various processes. The centre part of each 16-bit data is regarded as the column, while the corner bits are considered as the row, to obtain S-boxes results.
6. XOR & XNOR operation: Simple logical logic operations that are suitable for causing confusion and diffusion of outcomes. Both of these logical procedures are simple but necessary for the algorithm to function. XOR manipulates the data produced by the *f*-function as well as data swapped from the right side, whereas XNOR manipulates the key of each round as well as the data to confuse

3.3.5 The LSA flowchart

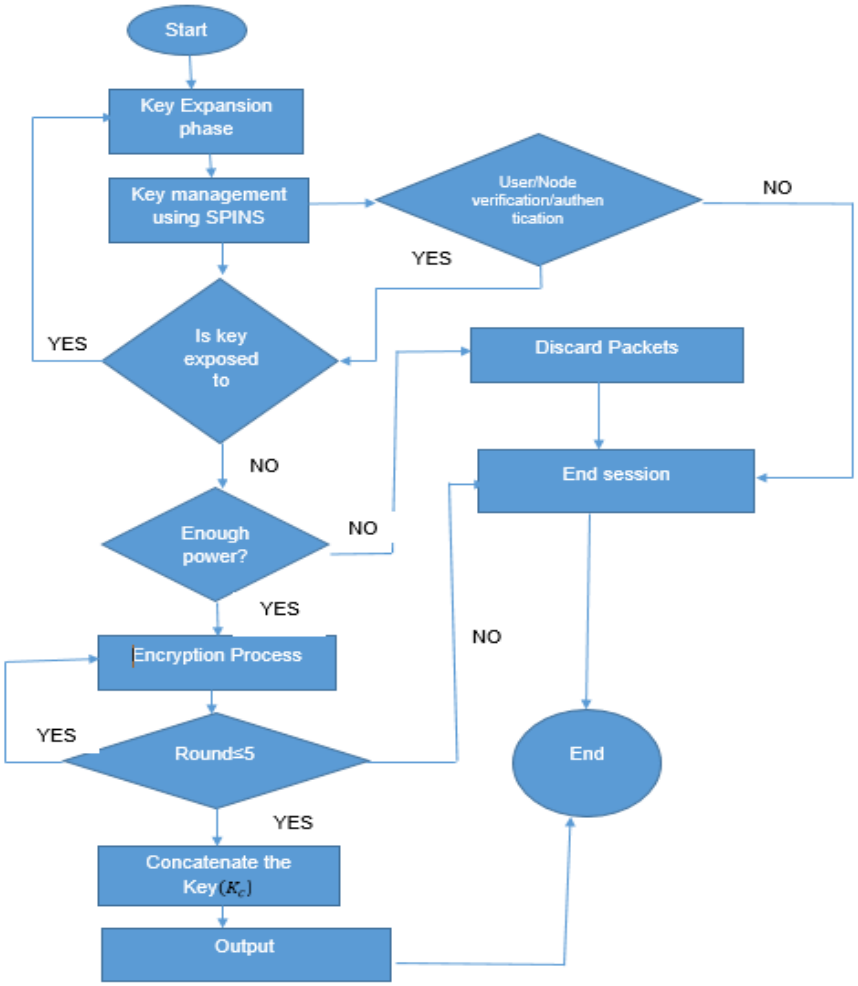

Figure 3.3: The LSA flowchart

3.3.6 The developed lightweight security algorithm

The three processes are carried out in order, one after the other, to ensure that users' sensitive data is protected and their privacy is preserved. The session is rejected if the answer comes from an unauthorized user or malicious node. The key will be discarded, and a new key will be requested from the key generation block. The proposed LSA, which integrates SIT encryption algorithm and SPINS security protocol to minimize security concerns in WSNs, is shown below.

Algorithm 3.1 Lightweight Security Algorithm

//key expansion

1. User inputs 64-bit Cipher key;
2. Divide 64 bit into 4 blocks 16-bit each;

3. Split a shift row is made for every 16-bits, output of shifting to feed to f-function;
4. Utilize four f-Function Block Cipher;
5. For i = 1 to 4
6. Perform an initial substitution of segments of cipher key (K_c) using equation (3.1)
7. Perform linear and non-linear modifications to generate confusion and diffusion using the P and Q table in Figure 3.2;
8. Construct key matrix KM1, KM2, KM3, and KM4. Each output of f-Function is made up of one matrix 4x4;
9. Matrix [4, 4] DataMatrix;
10. String (K_c);
11. If (key(K_c) length < 16)
12. Key round(K_r) are transformed;
13. Else if (key length > 16)
14. End session;
15. Next i;
16. End if;
17. The four round keys are the arrays formed ((K_1), (K_2), (K_3), and(K_4));
18. Perform XOR operation to obtain the fifth key using equation (3.3);
19. Combine Left and Right keys;
20. For K = 1 to 64;
21. If RM = 0 then –Cr-R [K] = 1, Cr-L [K] = 1;
22. Else Cr-R [K] = 0, Cr-L [KM] = 1;
23. End if;
24. End if;
25. Next K;

//key management
26. Verifying key (K_1, K_2, K_3, K_4, and K_5) and performing authentication user/nodes and key secrecy using Equation 3.4;

27. Check if K_{mac} for both N_s or N_r match;

28. Validate using (CBC-MAC) on$E_{(K_enc,C}(M)$;

39. Sensor validation=N_s or N_r;

30. For i= 1 to 5

31. Perform data freshness and integrity using Equation (3.5);

32. Check power on the nodes using Equation (3.6);

33. Calculate the power used on each round using Equation (3.7);

34. Calculating the sum of the power on the network using Equation (3.8);

35. If $E_{network}$ is more/equals to N_s or N_r power

36. Transmit data Packets;

37. Else

38. Discard data Packets;

39. Next i;

40. End if;

//encryption

41. Verify key (K_1, K_2, K_3, K_4, and K_5) receives

42. Perform *f*-function

43. If *f*-function =16

44. F = OR (S-boxes (AND (LS (16 bits/4) 16 Bit;

45. for i= 1 to 5

46. Perform Equation (3.9) to generate keys and for efficient power consumption;

47. next i;

48. Concatenate keys to obtain Cipher Text (Ct) from Equation (3.10);

49. End If;

50. End;

51. Output Ct;

3.4 Summary

This chapter presented the design of the proposed LSA by integrating SIT encryption algorithm and SPINS security protocol to provide data security; while minimizing computational complexity and power consumption to meet user demands and to provide secure connections. Transformation rounds containing a variety of transformation operations involving XOR, XNOR, *f*-functions, and swaps were used in the construction of the proposed LSA to create enough data confusion and diffusion to defeat attacks. S-Box is made up of multiple operations that mix the data and make the results more complex, making it impossible for the attackers to find the key because each round uses a different key.

The SPINS security protocol ensured that the secrecy of the key was met while checking the energy of the network and whether the nodes can transmit data without interruption. This was done to increase key sensitivity and accuracy while also minimizing complexity. The proposed algorithm, however, is limited to only five rounds to increase energy efficiency. To maintain the requisite security, the computation involved in the key expansion process must also be decreased.

Chapter 4: System Implementation

4.1 Introduction

The implementation of LSA is presented in this chapter. An IEEE 802.11s model generated in MATLAB version 2022a was used to implement the proposed LSA technique. On the computing platform, a Windows 10 computer with 8GB of RAM and an i5 Core CPU was deployed. In a 400*400 square metre space, 100 sensor nodes were randomly arranged with a radius of 20; the nodes distribution is random and clustered. Mesh routers are not typically placed nearby one another in the real world. As a result, the connection was set to 1 unit, and the minimum distance between nodes was set to 0.3 cm.

This chapter is organized thus: An overview of various wireless network simulation tools is presented in Section 4.2. The study compares various wireless network simulators in Section 4.3, and the LSA simulation is described in Section 4.4. In Section 4.5, the summary of the chapter is presented.

4.2 An Overview of Wireless Network Simulation

The network simulator is a discrete event packet-level simulator. The network simulator supports a wide range of protocols and network types, as well as different network features and traffic models. A network simulator is a collection of tools for simulating network activity, such as building network topologies, logging events that occur under various loads, analysing the events, and comprehending the network. The simulator requires both non-functional and functional-specific requirements for testing and evaluation (Vasanthi, 2017). Open source, platform independence, and a visualization module are non-functional requirements, while functional requirements include hardware simulation, battery, and power models, propagation modelling, protocol modelling, physical environment, and emulation.

Stochastic simulation, parallel simulation, continuous simulation, deterministic simulation, hybrid simulation, discrete event simulation (DES), and hybrid simulation are all examples of simulation. The simulation type of the simulator is mainly influenced by the type of application. Before the virtual implementation of a system that has not yet been

developed, simulation is performed to check for serious vulnerabilities or other design difficulties that may occur in the future (Sharma et al., 2020). In this research study, the MATLAB simulation tool was used because it is an open-source and DES.

4.2.1 Network simulators

To simulate communication in wireless networks, a variety of network simulators are used. This section discusses some of the most important network simulators, including the reason for choosing MATLAB for this research study. The simulators listed below were selected due to their broad use in the research community.

1. MATrix LABoratory (MATLAB)

MATLAB is written in the C and Java programming languages; MATLAB may be run on a variety of platforms. MATLAB comes with several tools and functions for analysing, visualizing, and measuring data. Filtering, signal processing, and signal modulation are examples. MATLAB is a tool for modelling, developing, and researching dynamic systems that use a graphical programming language. MATLAB is a simulation tool that comes with several toolboxes for creating new network scenarios (Sharma et al., 2020).

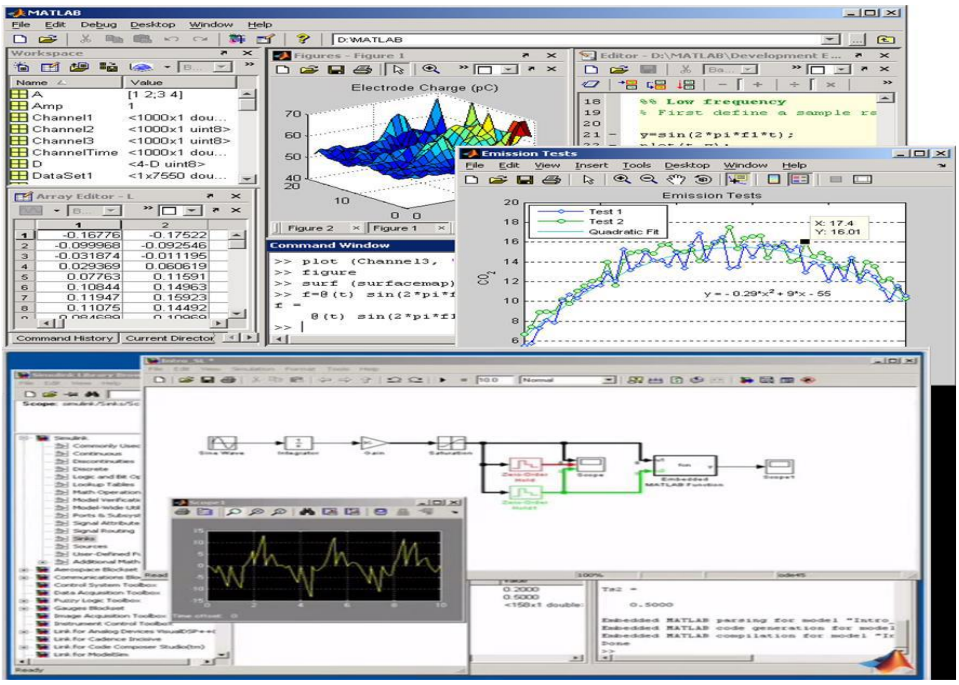

Figure 4.1: MATLAB simulator interface

2. Network Simulator-2 (NS-2)

NS-2 is an open-source DES tool that has become extremely popular in the modelling of communication networks. NS-2 was released in 1996. NS-2 supports a variety of protocols, including 802.11, 802.15.4, 802.16, and Impulse-Radio Ultra Wide Band (IR-UWB). The two languages that the object-oriented simulator NS-2 works with are named Object-oriented Tool command language (OTcl) and C++. These languages both play different roles in simulation modelling; for example, C++ is utilized to implement various protocols, whereas OTcl is used to define network topology, graphically present simulation outcomes, and create network scenarios (Vasanthi, 2017). One of the benefits of using NS-2 is the reduction of packets and event processing time. One of the drawbacks is that NS-2 is more complex and time-consuming at times for conducting the desired task than other simulators; and it does not have GUI.

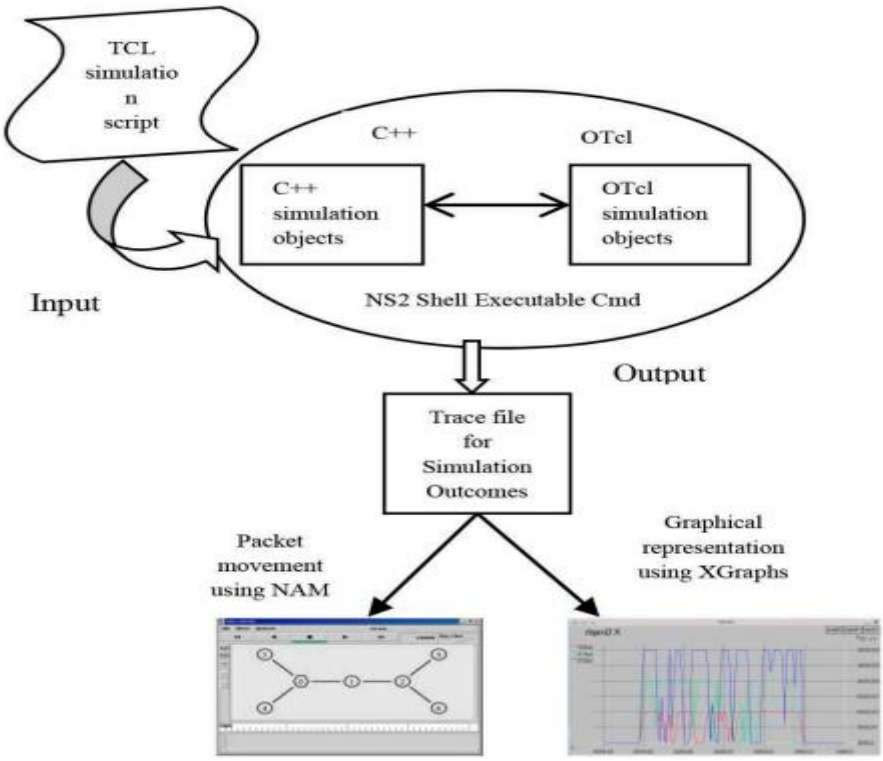

Figure 4.2: NS-2 Simulator interface

3. Network Simulator-3 (NS-3)

NS-3 is an open-source DES tool that was first released in June 2008. It is not an enhanced version of NS-2, and it does not support any of the NS-2 Application Program Interfaces (APIs). The NS-3 simulation language is pure C++ with optional Python bindings. The NetAnim software is used to provide graphical results. A simulation code can be written in either C++ or Python (Sharma et al., 2020). One of the benefits of using NS-3 is to allow modular libraries. One of the drawbacks of this simulation tool is that it suffers from a lack of credibility.

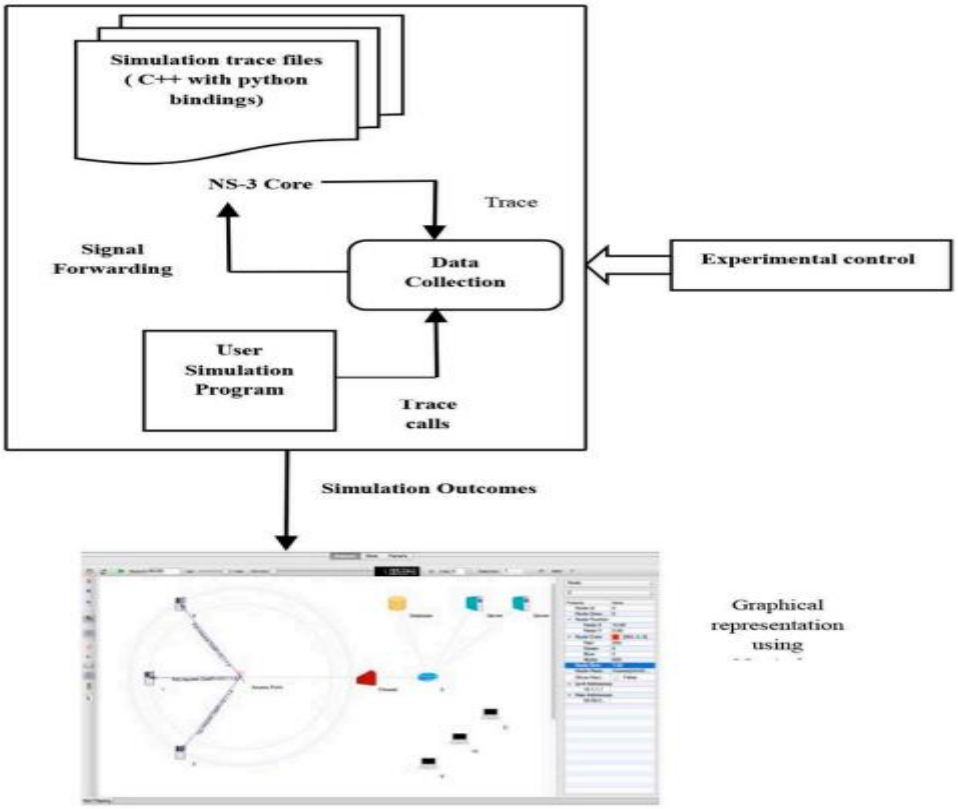

Figure 4.3 NS-3 Simulator interface

4. OPNET

OPNET is an object-oriented and discrete event simulator. OPNET is not a free and open-source simulator. OPNET is a commercial simulation suite with a wide variety of tools for creating network simulation environments via an interactive GUI. The GUI enables the users to model different types of network scenarios very quickly using drag-and-drop options. OPNET supports Windows, Linux, Sun Solaris, and high-level GUIs, with the majority of the programming achieved in C or C++ (Sharma et al., 2020).

OPNET is a high-level network-level simulator based on events. It was designed for the simulation of fixed networks and operates at the 'packet level'. OPNET has a huge collection of accurate simulations of commercially available fixed network hardware and protocols (Gnanaselvi, 2018). The simulator has a great deal of potential; however, there are not many contemporary wireless systems available. OPNET can be used for research as well as network design and analysis (end-user). For the developer, the usage threshold is high, while for the end-user, it is low. One of the benefits of OPNET is that it is a good combination of graphical and programmable interfaces. One of the drawbacks is that scalability problems are introduced due to the Object-oriented design of OPNET.

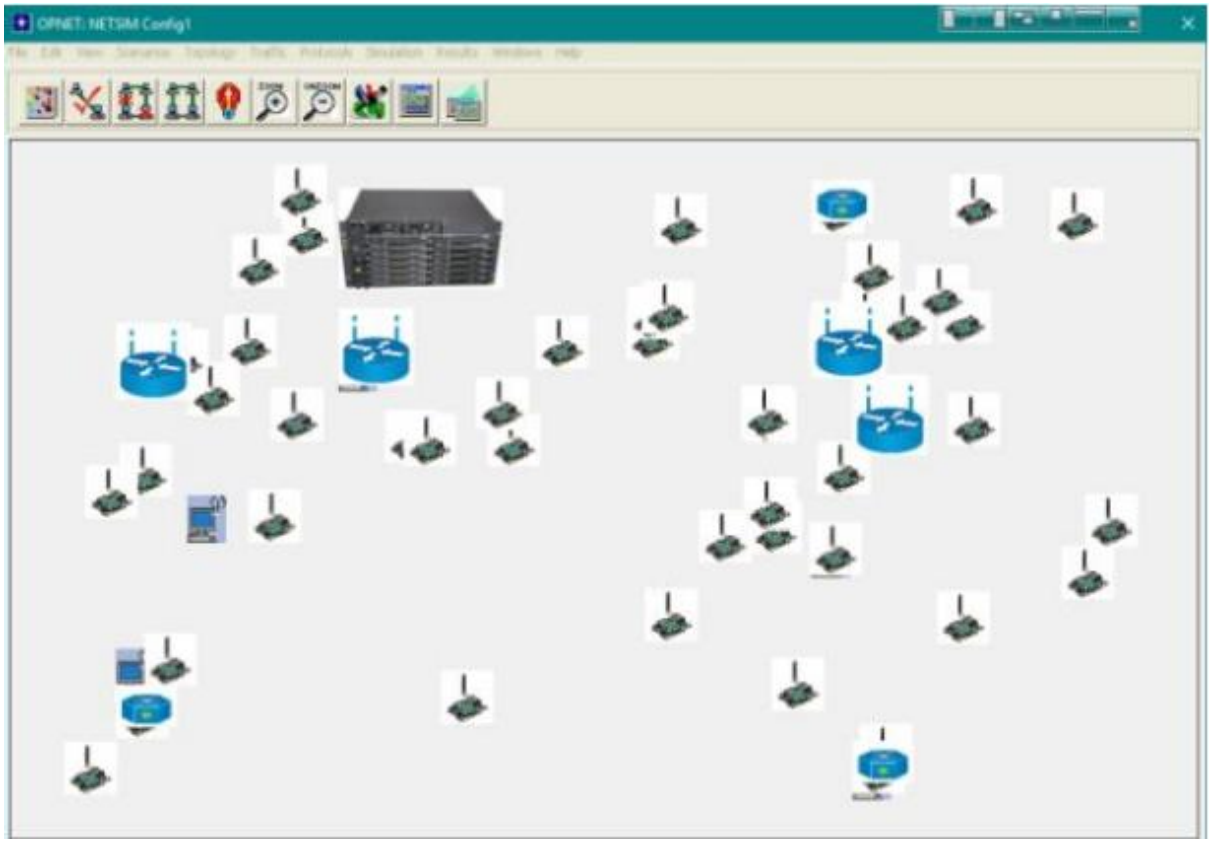

Figure 4.4: OPNET simulator interface

4.3 Comparison of Network Simulator

There is a criterion for choosing a good WSNs simulation tool. Table 4.1 compares the performance of various wireless network simulators (NS2, NS3, OPNET, and MATLAB). All

of the network simulators shown are accessible to use and may be accessed on the internet.

Table 4.1: Comparison of WSNs Simulation Tool based on Multiple Attributes

Simulator Name	**NS-2**	**NS-3**	**OPNET**	**MATLAB**
Simulation type	DES	DES	DES	DES
Programming Language	C++, Otcl	C++, Python	C,C++	C,C++,Java
Availability	Traditional model	Traditional model	Traditional model	Toolbox for data acquisition
Programmability	Strong	Strong	Strong	Very Strong
Requires Licence	Open Source	Open Source	Commercial	Open Source
Scalability	Small	Large	Medium	Very large
Operation System	Linux, Solaris, MAC OS X	Linux, Solaris, MAC OS X	Linux and Windows	Windows, Linux, and MAC OSX

The MATLAB simulator is the perfect tool of choice for the simulation environment because is a high-performance numerical computation and visualization software package. MATLAB is an interactive environment that includes hundreds of built-in functions for technical computations, animations, and graphics. The built-in functions of MATLAB are useful for linear algebra computations, signal processing, optimization, data analysis, and a variety of other scientific computations. MATLAB/Simulink provides all the qualities of an excellent simulator. MATLAB is a significant simulator that can simulate a huge number of sensor nodes. MATLAB has a full set of toolboxes, built-in functions, and algorithms. It provides a high level of simulation accuracy and can capture the behaviour of dynamic

systems at a fine-grained level. Table 4.2 shows the features and functions that make MATLAB the best simulation tool.

Table 4.2: Features and Function for MATLAB Simulation (Sharma et al., 2020)

Features	Properties
Simulink coder	Helps in the automatic generation of C-based code for systems implementation
Embedded coder	Provides support for embedded systems modelling
HDL coder	Automatically outputs hardware description languages Verilog, and VHSIC hardware description language
Stateflow	Helps in designing diagrams, state machines, and flowcharts
xPC target	Simulates and tests state-flow models in real-time
SimEvents	A simulation library of graphical building tools for modelling systems

4.4 Simulation for Lightweight Security Algorithm

The key objective of simulating the LSA was to decide whether the proposed algorithm enhances the security of WSNs in the IoT. In this section, the details on how to configure MATLAB in Windows are shown and discussed. The MATLAB configuration interface is shown in Figure 4.5.

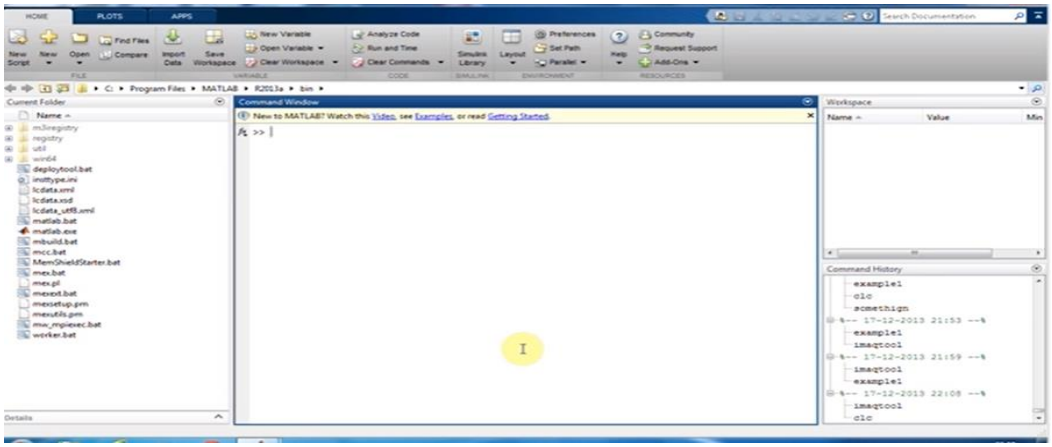

Figure 4.5: MATLAB configuration interface

MATLAB is a very high-performance simulation tool, which is why the simulation tool was chosen for this research study. MATLAB 2022a is the most recent version currently available for download. A simulation example is presented to demonstrate how to configure MATLAB on a Windows 10 system, as described in Section 4.1. The simulation topology is shown in Figure 4.6, as configured in MATLAB.

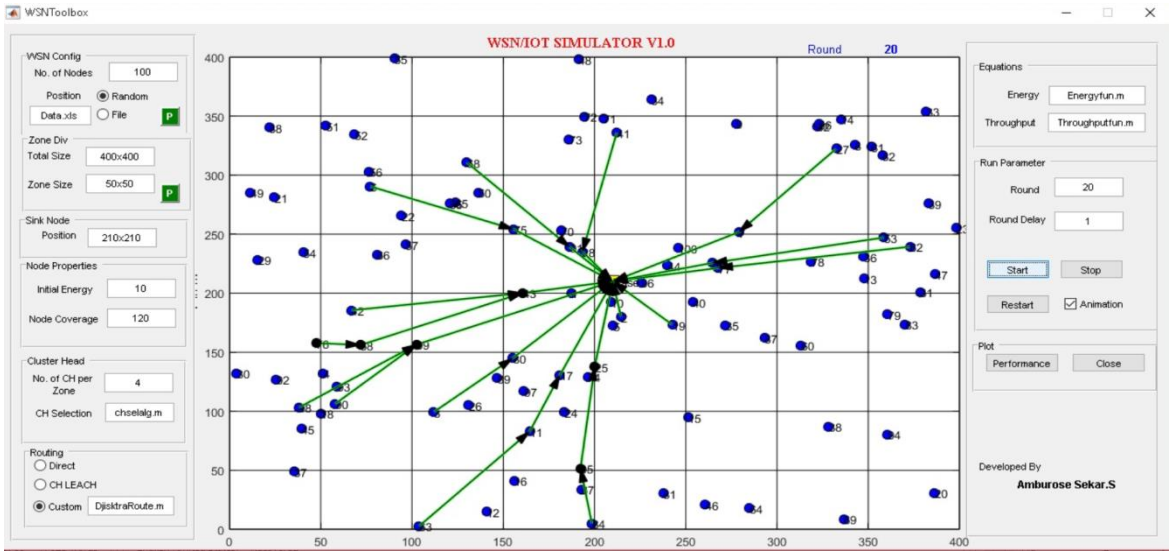

Figure 4.6: Configuration parameters

The tool MATLAB was chosen primarily because it was created with design efficiency parameters in consideration. MATLAB is efficient in numerical computations, powerful in modelling, and simple to use. Table 4.3 shows the simulation parameter settings for the algorithm.

Table 4.3: Simulation Parameters

Parameters	**Values**
Simulator	MATLAB 2022a
Protocol	IEEE 802.11s
Packet size	600 bytes
Simulation area	400mx400m
Mobility mode	Random

Channel type	Wireless channel
Type of network interface	Wireless
Number of nodes	100
Simulation time	100s
Number of rounds	5
Connection radius	1 unit
Execution radius	20

The IEEE 802.11s protocol was chosen for this study because it allows the user to model communication between several WLAN nodes that contain both the MAC and physical layers. Many IoT protocols rely on IEEE 802.11s, because of a low-data-rate wireless personal area network. To build a complete WSNs system, in this research study we employed the MATLAB Simulink communication block set. The simulation technique includes creating a hardware design for the transmitting of nodes, as well as receiving master node architecture and modelling the communication channel. Establishing safe data transmission and key establishment protocol allows for interworking and security; the research study aims to improve these aspects.

4.5 Summary

In this chapter, the implementation and simulation of the proposed LSA algorithm using MATLAB are discussed. The chapter included a comparison of wireless network simulators and a detailed discussion of the MATLAB simulator, including the features that made it suitable for use in this research study. The simulation results obtained from numerous simulations undertaken to evaluate the performance of the proposed LSA algorithm are presented in the next chapter.

Chapter 5: Results and Discussion

5.1 Introduction

A simulation is useful because it allows the user to see how efficient the proposed algorithm is before putting it into practise in a real-world setting. Researchers can use a network simulator to evaluate the performance of a network under diverse circumstances. The simulator can be tweaked to produce more accurate outcomes for the analysis. This chapter uses graphical drawings to illustrate and analyse the results of the simulation used to evaluate the efficiency of the LSA in a range of network setups. The Xgraph was used to display the MATLAB results that were evaluated.

This chapter is structured as follows: In Section 5.2, the study presented the experimental evaluation of the proposed LSA by comparing it with the commonly known algorithms named SPN and Feistel. Thereafter Section 5.3 presents simulation findings and analysis before concluding the chapter in Section 5.4.

5.2 Experimental Evaluation

To measure the effectiveness of the proposed LSA in WSNs in IoT, experiments were provided. In the IEEE 802.11s protocol, 100 random nodes were deployed, each capable of running up to 5 rounds. To obtain more convincing results, each simulation was run for 100s. The space between nodes must be at least 0.3 cm. To allow communication between numerous wireless network nodes, the IEEE 802.11s protocol was used. The encryption process was carried out via LSA encryption, with various keys known to the receiver.

5.3 Simulation results

To monitor the effectiveness of the encryption techniques, a simulated scenario was used, as stated in Section 4.1. The simulation results in this study were compiled from an average of 10 simulations. The simulations' performance indicators are focused on the following parameters:

1. Energy consumption: the overall energy consumed by the network to execute data transmission, data processing, and reception, either between the sensor nodes or

between sink nodes and sensor nodes. One of the most important elements of WSNs-IoT-based environmental monitoring is energy usage. Attackers will not then be able to manipulate or alter data when the network is out of power, or should the power be insufficient.

2. Security: One of the most critical key metrics in network communication is security. If security is breached, the entire network may become vulnerable to a variety of attacks. The greater the key length and the higher number of rounds, the more robust the security. WSNs in the IoT are used in a variety of critical areas, with security a major factor.
3. Packet Delivery Ratio (PDR): This is the ratio of the number of transmitted packets by the source to the number of received packets by the destination. Data collisions, intermediary node failures, forced path switching, excessive data traffic, and intruder attacks are all examples of PDR.
4. Key expansion time: is the time required to execute the key expansion process; the shorter the key expansion time, the more efficient the algorithm. Users with unauthorised access will not have sufficient time to access the key; therefor e the secrecy of the key will be protected.

The evaluation of the effects of energy consumption, security, PDR, and key expansion time on network performance is presented in this section. The proposed algorithm has been compared with well-known cryptographic algorithms such as SPN and Feistel, for effective results. The SPN and Feistel algorithms are both lightweight block ciphers suitable for resource-constrained devices such as sensor nodes. Both algorithms use S-Box and mathematical operations for the encryption process. However, for better security, Feistel requires more rounds, and the more rounds, the higher the power consumption. Furthermore, network performance will suffer as a result. The SPN has more 'inherent parallelism', encryption, and decryption can be completed faster.

5.3.1 Security

One of the most critical key performance indicators in network communication is security. If security is jeopardized, the entire network may be at risk. Security is the most critical aspect to consider, especially in the context of WSNs in the IoT utilized in a sensitive environment.

The size of the key is very important when it comes to confidentiality and secrecy level. There should be no faster attack than an exhaustive key search for a secure block cipher. Because exhaustive key search takes much longer for a larger key size, a theoretical attacker may afford to put in more effort to 'crack' the larger cipher. As illustrated in Figure 5.1 for key lengths from 8-bits to 256-bits during data transmission, LSA maintains a security level of 97% to 99%. After the key is generated, the key is validated by the key-management process. Validation ensures that the secrecy of the key is sufficiently strong, and therefore not exposed to any kind of attacker. In addition, only authorized users with authorization can access the data in transit. To protect the confidentiality and secrecy of data, any key length can be utilized with a significant number of rounds. LSA has five rounds indicating higher security against cryptanalysis because there is more confusion and diffusion.

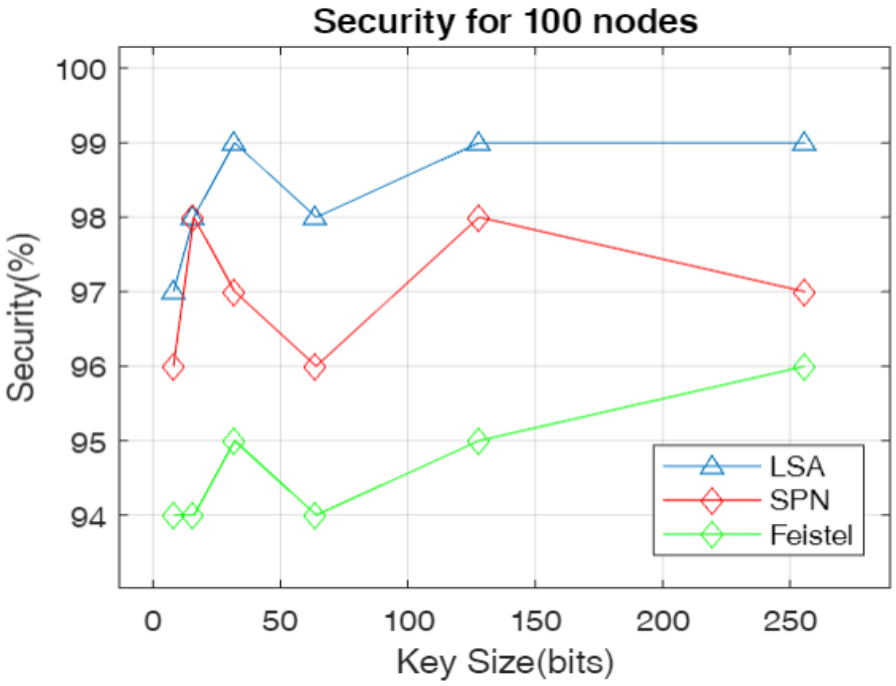

Figure 5.1: Security for 100 nodes

SPN was the next-best performer, with a level of security for 8-bits to 256-bits ranging from 96% to 98%. Feistel was the weakest performer with a security range of 94% to 96% for 8-bits to 256-bits. However, the SPN and Feistel do not have the key management process to authenticate the users. Neither SPN nor Feistel cannot check whether the key has been exposed to any attacks nor can they validate the key, which is why the secrecy of the key is seen to be at risk. This increase the chance of attack, such as key-related attacks. The secrecy

of the key is one of the most important aspects. Should the attackers have access to the key the security of the data, especially the integrity and confidentiality of the key, will easily be compromised. In all 10 simulations, the proposed LSA has outperformed the other techniques in terms of security.

5.3.2 Energy consumption

One of the most critical factors, when there are a huge number of battery-powered network nodes, is energy consumption. Energy consumption is a critical metric because IoT-based WSNs nodes are generally battery-operated devices. If there is insufficient power on the network the data transmitted becomes more prone to data theft, threat, and misuse. Energy is measured using Micro-Joule (uJ). In all 10 simulations, LSA used less energy than other techniques, as seen in Figure 5.2.

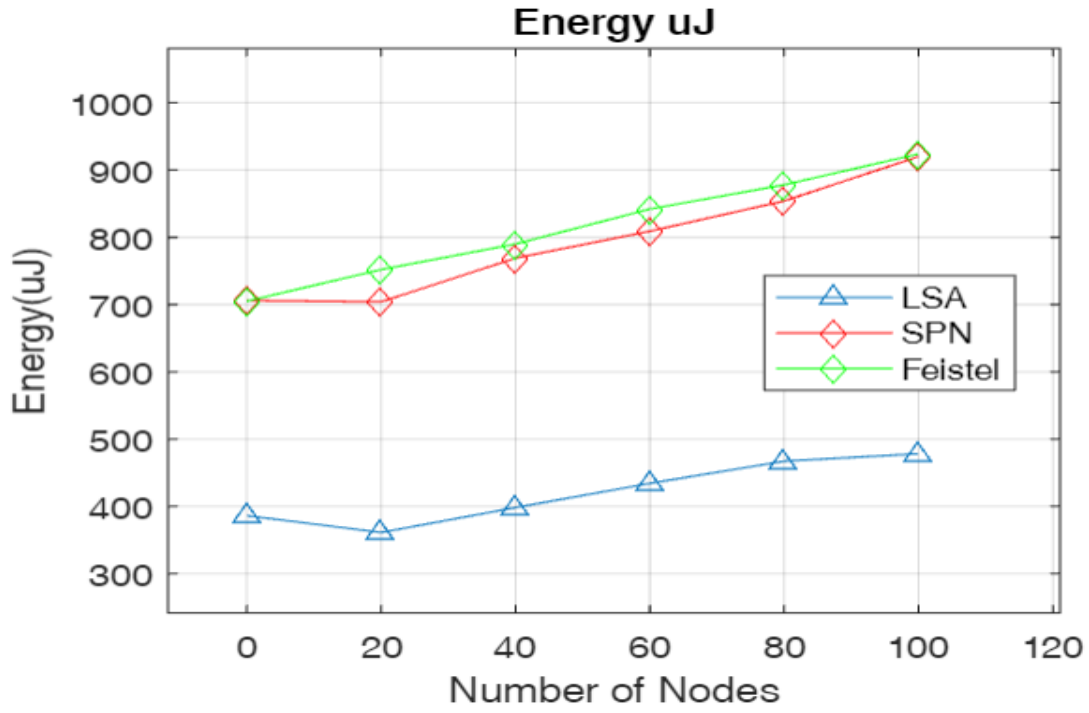

Figure 5.2: Energy for 100 nodes

LSA consumed 386uJ for data transmission when simulated with 10 nodes. When 100 nodes are utilized, the highest energy used by LSA was 478uJ. LSA's average energy usage was 411.2uJ, which is one of the lowest amounts compared with other techniques. SPN showed an average energy consumption of 781.6uJ, and therefore is rated the runner up. However, SPN uses complicated mathematical procedures during key expansion. The number of rounds

generated is made up of key mixing, the substitution layer, and the permutation layer that increases power usage. Feistel requires many rounds for robust data security and for ensuring the strength of the key secrecy, it has a higher power utilization of 705uJ to 924uJ for 10 to 100 nodes. The average energy consumption is 795.3uJ for all 10 simulations. The fewer the number of rounds, the lower the energy consumption. The proposed algorithm also checks whether the network has sufficient power for the data to be transmitted, to avoid data loss. The results showed that LSA energy consumes less energy compared with the SPN and Feistel.

5.3.3 Packet delivery ratio

PDR is defined as the ratio between the packets sent from the source node to packets successfully delivered to the destination node. The high rating of PDR indicates minimal packet loss and data collisions. The proposed LSA has shown an average of 94.5% PDR. This is because the PDR was maintained between 90% and 99% when using the proposed LSA. The second-best PDR results were obtained by SPN, with a minimum of 78% and a maximum of 95%, thus an average of 86.5%. The worst-performing algorithm is Feistel, with a minimum of 75% and a maximum of 96%, thus an average of 85.5%.

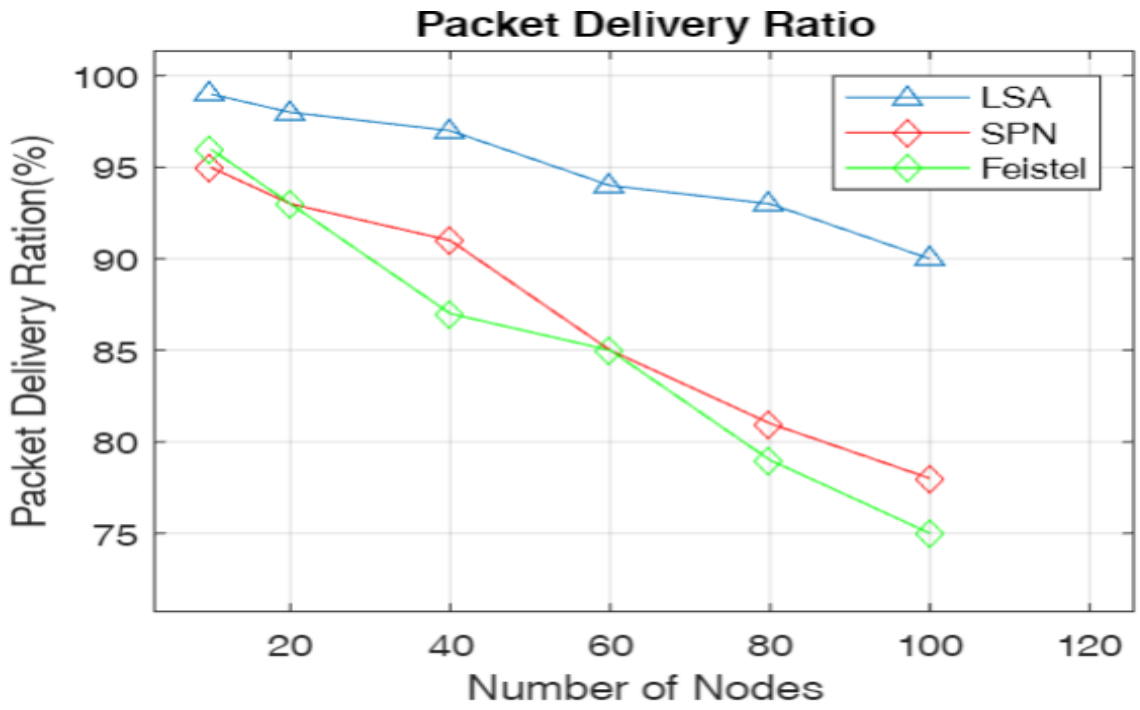

Figure 5.3: Packet delivery ratio for 100 nodes

The simulation results showed that the proposed LSA is more fault-tolerant when it comes to security threats. The highest PDR symbolizes the low number of packet drops and the low number of data collisions. Thus, the number of possible attacks on the data transmitted in WSNs in the IoT was reduced.

5.3.4 Key expansion time

Figure 5.4 illustrates a comparison of the lightweight encryption techniques' key expansion times. The bit length determines the key expansion time, which is the time it takes to generate a key. The greater the length, the longer the time. The findings indicate that LSA used less time than SPN and Feistel. LSA took a minimum time of 50mS on a key length of 8-bits to generate the key; and a maximum of 102ms on a key length of 256-bits to generate the key. The Feistel comes in second with a maximum of 115mS on a key length of 256-bits.

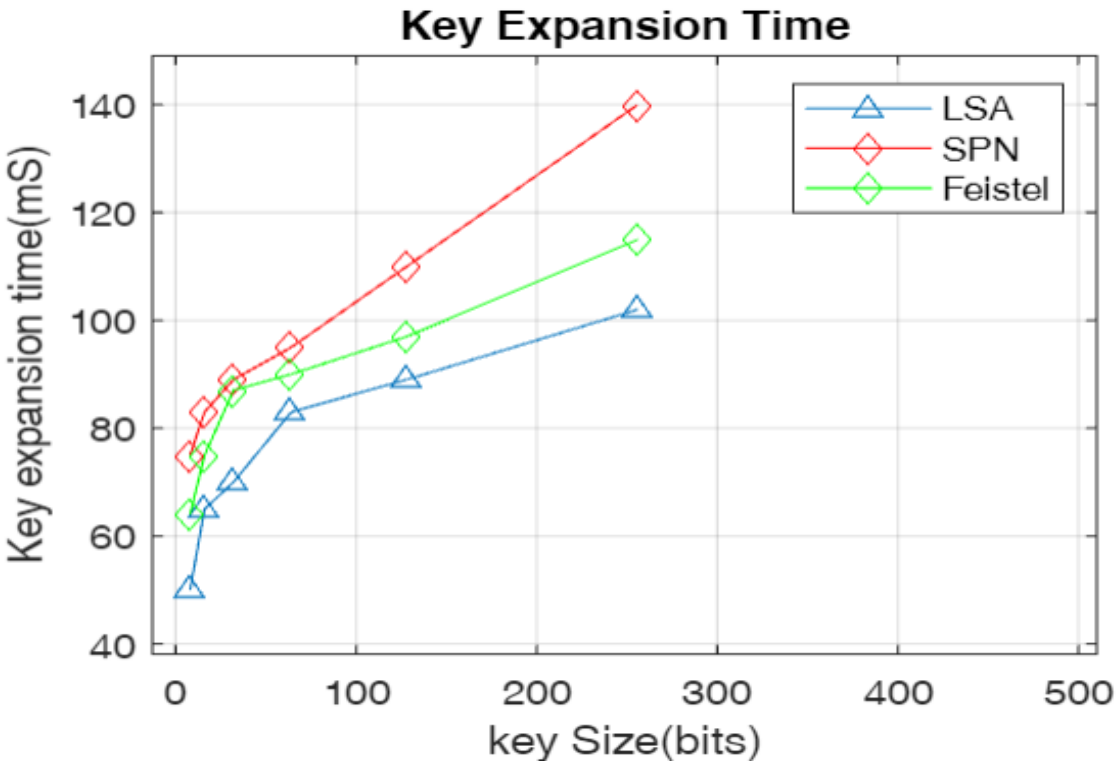

Figure 5.4: Key expansion time (LSA, SPN, and Feistel) for 100 nodes

As illustrated in Figure 5.4, when the key length is 8-bits, the SPN took a minimum of 75mS to generate the key. For a maximum of 256-bits key length, it took 140mS to generate the key. Because SPN has a complicated and longer key expansion process that uses mathematical operations and because the time taken to execute is very long, the secrecy of the key is at

risk. The attackers have sufficient time to easily manipulate or alter the key and resend it. If the execution time is longer, this will affect the security of the data. The LSA outperformed both SPN and Feistel; thus, the proposed LSA provides better data security with a shorter key expansion time.

The findings demonstrate that the LSA has enhanced data security and flexibility. The LSA has a straightforward architecture that contains five rounds of encryption, each of which requires simple mathematical procedures. This reduced the computational complexity and processing resources. The dynamic key generation technique was carried out at the encryption to reduce the encryption strain. Even though the number of mathematical operations per round and the use of S-Box in LSA are higher, the overall complexity is lower. Most mathematical operations use 4-bit data; however, S-Box come in only 4x4 sizes. Other mathematical techniques that work with 8 or 16-bits of data differ in this way. The proposed algorithm's data-processing time and key-generation time are both faster than those of other techniques with secure key generation.

This methodology allows users to obtain an acceptable level of security and rapid data processing efficiency when using the proposed LSA. The proposed LSA increased the security and integrity of data transferred in WSNs in the IoT while making it available on demand. The LSA can withstand the most well-known attacks, such as linear cryptanalysis, related-key attacks, differential, weak-key attacks, and symmetric properties.

Finally, a brief description of the LSA technique is presented based on the data encryption enforcement properties of WSNs in the IoT.

1. Security: LSA is a secure algorithm because it uses a complicated structure and a combination of SPN and Feistel architectural techniques with the security protocol.
2. Key expansion process: LSA offers an effective key procedure that helps to avoid key-related attacks due to the *f*-function and matrix expansion of a key rather than a single extension key. The security level of LSA will be enhanced due to the adoption of the key expansion technique. In addition, the use of the security protocol for key validation also increases the level of data security and lowers the chances of possible attacks.

3. Power consumption: LSA uses only a limited number of rounds. The proposed LSA technique is therefore superior when it comes to balancing power consumption and data security. If the network has sufficient power the chances of attackers manipulating data and packet loss are very slim.
4. Time complexity: Because the need for subsequent rounds has been reduced, there is no additional issue in terms of time.

5.4 Summary

The findings of the simulation scenario discussed in Section 4.1 were provided in this chapter. The study also considered the effectiveness of lightweight encryption techniques in WSNs in the IoT. The following network factors were used to present the results: energy consumption, security, PDR, and key expansion time. The LSA algorithm, SPN, and Feistel algorithms were compared using the MATLAB simulation tool to analyse findings and compare performance. In this chapter, the proposed LSA was proven to be capable of reducing power consumption while improving data security without impacting network efficiency. In terms of PRD, key expansion time, security, as well as preserving power consumption, the LSA outperformed the well-known approaches such as SPN and Feistel.

Chapter 6: Conclusion and Future Work

6.1 Introduction

In this study, an LSA was developed by integrating the SIT and SPINS to enhance the data security of WSNs in the IoT without sacrificing network performance. The proposed LSA reduced power consumption and computational complexity, the number of attacks, and improved data security.

The rest of the chapter is organized as follows: In Section 6.2, the research study discusses how the technical objectives were met. The overall summary of the research is presented in Section 6.3. The obstacles encountered throughout this study are reviewed in Section 6.4. Future work is suggested in Section 6.5, and the chapter conclusion is presented in Section 6.6.

6.2 Achievement and Objectives

The objectives discussed in Section 1.6 of Chapter 1 have all been achieved accordingly.

1. **To identify the existing encryption algorithm for WSNs in the IoT**

   The literature review in Section 2.6 identified and analysed the existing security approaches used for WSNs in the IoT. The benefits and limitations of the SPN, Feistel, and hybrid solutions, which use symmetric and asymmetric algorithms, were discussed. The SPN procedures are often carried out via XOR/bitwise rotation, with sections of the key entered as the rounds advance. After that, the encryption rounds are reversed, and the S-boxes/P-boxes function in the opposite direction.

   However, the SPN has more ‘inherent parallelism’, making encryption and decryption readily completed. With Feistel, the main advantage of this technique is that encryption and decryption are nearly equivalent, if not identical in some cases, with simply a key schedule reversal required. Furthermore, the round function in Feistel networks does not have to be invertible. The main drawback is that Feistel requires too many rounds and the more rounds the higher the power consumption.

Because most sensor nodes are battery-powered, power limitations are the key concern for WSNs in the IoT. Thus, the proposed algorithm minimized power consumption. The proposed LSA's main advantage is to integrate the features of the two algorithms named SIT and SPIN to improve data security and encryption throughout execution, as well as to decrease cryptanalytic vulnerabilities.

2. **To design the proposed LSA encryption algorithm by integrating SIT and SPINS in WSNs in the IoT**

A discussion of how the proposed LSA was developed was presented in Chapter 3. The SIT and SPINS techniques were integrated to reduce the number of attacks and power consumption while improving data security without impacting network performance. The use of SIT was important in this research study, as it allowed for the creation of faster, smaller, and more efficient cryptographic keys.

By achieving data confidentiality, integrity, and computational complexity, the SIT benefitted this research study more in its key expansion and encryption phase. During the key management phase, the SPINS assisted in a high level of security with lower power consumption and authenticating users. Before the encryption procedure, the SPINS ensured that the key was secure and that the time it took for a user to be authenticated in the network was kept to a minimum. The modified path for carrying out all steps involved in the LSA was illustrated using a flowchart, as shown in Figure 3. MATLAB simulator software was used to implement the proposed LSA as described in Chapter 4.

3. **To evaluate the effectiveness of the proposed LSA against existing encryption algorithms**

Security, energy consumption, PRD, and key expansion time were parameters used to evaluate the performance of an LSA against SPN and Feistel in WSNs in the IoT. To record the results, Simulink blocks were used to evaluate and view efficiency gains. The analysed results were graphically displayed using an x,y graph called Xgraph, in which the behaviour of the LSA was compared with commonly known cryptographic techniques, namely SPN and Feistel. When compared with SPN and Feistel, the proposed LSA performed significantly better. For all five rounds, the proposed LSA maintained a security level of 98% to 99% for

key sizes ranging from 8 to 256-bits. When compared with SPN and Feistel, LSA's average energy usage was 411.2uJ, which is one of the lowest compared with Feistel and SPN. Using LSA, the PDR was maintained between 90% and 99% over SPN and Feistel in just 10 simulations. On average, LSA has an average of 94.4% PDR. The key expansion time during the key expansion process was reduced to 102 ms for LSA comparing it with Feistel and SPN. Thus, the proposed LSA outperformed both techniques. The results showed that the time taken to process and execute the key during the key expansion process was minimal when LSA was applied by comparison with the SPN and Feistel. As a result, the proposed LSA minimised the number of attacks, hence producing better data security by reducing power consumption compared with SPN and Feistel.

6.3 Summary of Research

During data transmission within the network, the proposed algorithm reduced packet loss while satisfying the power consumption limitation, boosting data security, and retaining network performance. MATLAB simulation software was used to build and execute the proposed technique. The study presented the design of the proposed LSA to improve data security for WSNs in the IoT. The proposed LSA was designed by integrating SIT and SPINS in order to minimise power consumption, reduce the number of attacks, and enhance data security during data transmission without affecting network performance. Security requirements such as authentication, integrity, and confidentiality were fulfilled. The simulation results showed a reduction in key expansion time, together with the amount of packet loss, and power consumption, as well as an enhancement in data security when the proposed LSA was compared with SPN and Feistel.

6.4 Challenges Encountered

Because the MATLAB network simulator supports different languages, translating the proposed technique and implementing it was a tricky task. Gaining the license for MATLAB was not straightforward; however, using my tut4life email address was the solution, MATLAB granting access to institutions. The design of the proposed algorithm, as well as learning the MATLAB language, were also challenges. However, YouTube and MATLAB videos helped me to understand and overcome all challenges.

6.5 Future Work

The implementation of the technique on software and hardware in various computational and network environments is being considered for future research. The LSA has shown a substantial improvement in data encoding, offering high security and moderate computational complexity, and low power consumption, according to experimental data. The proposed LSA proved to be much more useful for WSNs in the IoT environment because of its quick data collection, data confidentiality, and data integrity. In the future, the proposed LSA can be tweaked to improve the execution time of the encryption phase. By modifying the number of rounds, or by altering the architecture to allow for different key lengths, the scalability of the algorithm can be utilized for greater security and speed. The LSA method could be implemented in hardware in the future, resulting in significantly better results.

6.6 Summary

This chapter has established that the proposed LSA is capable of reducing power consumption while improving data security without affecting the performance of the network. In terms of PRD, key expansion time, and security, as well as preserving power consumption, the LSA provided robust security.

References

AAKASH, D. & SHANTHI, P., 2016. Lightweight Security Algorithm For Wireless Node Connected with IoT. *Indian Journal of Science and Technology*, 9(30):1-8.

AHMED, S. F., ISLAM, M. R., NATH, T. D., FERDOSI, B. J., HASAN, A. S. M. T., 2019. G-TBSA: A Generalized Lightweight Security Algorithm for IoT.4th International Conference on Electrical Information and Communication Technology (EICT)*:*1-6.

ALOTAIBI, M., 2019. Security to wireless sensor networks against malicious attacks using Hamming residue method. *Journal on Wireless Communications and Networking*, 8(2):1-7.

ANANDHAVALLI, A. & BHUVANESWARI, A., 2019. Lightweight Security Algorithm on PMIPV6 Protocol for IOT Based Wireless Sensor Networks. *International Journal of Innovative Technology and Exploring Engineering (IJITEE),* 8(12):1790-1799.

ANWAR, N.B., & MAHA, M.M., 2020. AMPC: A Lightweight Hybrid Cryptographic Algorithm for Wireless Sensor Networks. *International Journal of Innovative Science and Research Technology,* 5(6):1142-1146.

BANGASH, Y.A, ABID, Q.D., ALSHREEF, A.A.A, YAHYA, E. & AL-SALHI, A., 2017. Security Issues and Challenges in Wireless Sensor Networks: A Survey. *International Journal of Computer Science,* 44(2): 94-108.

BURHANUDDIN, M. A., MOHAMMED, A. A., ISMAIL, R., HAMEED, M., KAREEM, A. N.& BASIRON, H., 2018. A Review on Security Challenges and Features in Wireless Sensor Networks: IoT Perspective. *Journal of Telecommunication, Electronic and Computer Engineering,* 10(7):17-21.

CHATTERJEE, R. and CHAKRABORTY, R., 2020. A Modified Lightweight PRESENT Cipher For IoT Security, *2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)*:1-6.

CHELLI, K., 2015. Security Issues in Wireless Sensor Networks: Attacks and Countermeasures. *Proceedings of the World Congress on Engineering,* 1(2):519-524.

DENER, M., 2014. Security Analysis in Wireless Sensor Networks. *International Journal of Distributed Sensor Networks*:1-9.

EBRAHIM, M. & CHONG, C.H., 2015. Secure Force: A Low-Complexity Cryptographic Algorithm for Wireless Sensor Network (WSN).2013 IEEE International Conference on Control System, Computing and Engineering:557-562.

ENGELS, D.W., SAARINENE, M.J.O., SCHWEITZER, P. & SMITH, E.M., 2011. The Hummingbird-2 Lightweight Authenticated Encryption Algorithm:19-31.

GNANASELVI S., 2018. A Study on Various Simulation Tools for Wireless Sensor Networks. *International Journal of Engineering Research and Management (IJERM)*, 5(6):1-3.

HAMZA, R., YAN, Z., MUHAMMAD, K., BELLAVISTA, P. & TITOUNA, F., 2020, "A privacy-preserving cryptosystem for IoT E-healthcare", *Information Sciences*:493-510.

HUANSHENG, N., LIU, H. & LAURENCE, T., 2015. Aggregated-Proof Based Hierarchical Authentication Scheme for the Internet of Things. *IEEE Transactions,* 26(3):657-667.

JACINTO E.G., HOLMAN, M.A. & FERNANDO, M.S., 2017.Implementation of the cryptographic algorithm “Present” in different microcontroller type embedded software platforms, *International Journal of Applied Engineering Research*, 12(19): 8092-8096.

JAN, M., NANDA, P., USMAN, M. & HE, X., 2017. PAWN: a payload based mutual authentication scheme for wireless sensor networks. Concurrency and Computation: Practice and Experience, 29(17):1-15.

JAIN, A. & SINGH, T., 2019. Securing Communication in IoT Ecosystem Using Cryptographic Algorithms. *International Journal of Engineering and Advanced Technology (IJEAT),* 9(1):7258-7268.

KHARA, S., 2017. A Review on Security Issues in Wireless Sensor Networks. *International Journal of Sensor Networks and Data Communication*:2090-4886.

KHEMISSA, H., TANDJAOUI, D. & BOUZEFRANE, S., 2017. An Ultra-Lightweight Authentication Scheme for Heterogeneous Wireless Sensor Networks in the Context of Internet of Things. *In International conference on mobile, secure, and programmable networking*:49-62.

MAHAJAN, P. & SARDANA, A., 2017. Key Distribution Schemes in Wireless Sensor Networks: Novel Classification and Analysis. *In Advances in Computing and Information Technology*:43-53.

MAITRA, S. & YELAMARTHI, K., 2019. Rapidly Deployable IoT Architecture with Data Security: Implementation and Experimental Evaluation Sensors, 19(11):1-22.

NDIA, J.G., 2017.A Survey of Security Protocols for Wireless Sensor Networks. *International Journal of Applied Computer Science (IJACS),* 1(2):1-11.

OTHMAN, M.F. & SHAZALI, K., 2019. Wireless Sensor Network Applications: A Study in Environment Monitoring System. *International Symposium on Robotics and Intelligent Sensors*, 41(12):1204-1210.

PEI, C., XIAO, Y., LIANG, W. & HAN, X., 2018. Trade-off of security and performance of lightweight block ciphers in Industrial Wireless Sensor Networks. *EURASIP Journal on Wireless Communications and Networking:*1-18.

PIRBHULAL, S., ZHANG, H. E., ALAHI, M.E., GHAYVAT, H., MUKHOPADHYAY, S.C., ZHANG, Y.T. & WU, W., 2017. A Novel Secure IoT-Based Smart Home Automation System Using a Wireless Sensor Network,17(69):1-19.

PORAMBAGE, P., 2018. Lightweight authentication and key management of wireless sensor networks for Internet of things.*jultika.oulu.fi.* [Online]

Available at: http://jultika.oulu.fi/Record/isbn978-952-62-1995-0

[Accessed 04 May 2021].

POONGAVANAM, N., LATHAPARTHIBAN, P., VADIVELU, K., SITHINYAMOORTHI, H., & BALAJI, N., 2018. A Survey on Attacks and Security Goals in WSN. *International Journal of Pure and Applied Mathematics*, 119(14):1641-1646.

SABEEL, U., MAQBOOL, S., & CHANDRA, N., 2016. Categorized Security Threats in the Wireless Sensor Networks: Countermeasures and Security Management Schemes. *International Journal of Computer Applications,* 64(16):19-28.

SEN, J., 2019, Security in Wireless Sensor Networks. *Department of Computer Science & Engineering, National Institute of Science & Technology*. INDIA.

SHARMA, A., MARRIBOYINA, V. & GUPTA, K., 2019. Design and Implementation of key Generation Algorithm for Secure Image. *International Journal of Recent Technology and Engineering (IJRTE),* 8(2):5139-5146.

SHARMA, R., VASHISHT, V. & SINGH, U., 2020. Modelling and Simulation Framework for Wireless Sensor Networks: a comparative study. *The Institution of Engineering and Technology (IET)*, 10(5):181-197.

SHANTHA, M. J. R. & AROCKIAM, L., 2019. SAT_Jo: An Enhanced Lightweight Block Cipher for the Internet of Things,*2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*:1146-1150.

SINGH, G., 2016. Security Attacks and Defense Mechanism in Wireless Sensor Network: A Survey. *International Journal of Innovative Science, Engineering and Technology (IJISET)*, 3(4):129-136.

SINGH, S., SHARMA, P., MOON, S. & PARK, J., 2017. *Advanced lightweight encryption algorithms for IoT devices: survey, challenges, and solutions.* [Online]

Available at: https://doi.org/10.1007/s12652-017-0494-4

[Accessed 04 May 2021].

SWITZERLAND, 2014. Internet of Things: Wireless Sensor Networks.*www.scribd.com.* [Online]

Available at: https://www.scribd.com/document/374039000/iecWP-internetofthings-LR-en-pdf?cv=1, [Accessed 04 May 2021].

TANG, D., LI, T., RAN J & WU, J., 2015. Cost-Aware SEcure Routing (CASER) protocol design for wireless sensor networks. *IEEE Trans*. Parallel. Distributed Syst, 26(4):960-973.

TAWALBEH, H., HASHISH S., TAWALBEH, L. & ALDAIRI, A., 2017.Security in Wireless Sensor Networks Using Lightweight Cryptography. *Journal of Information Assurance and Security*, 12(1):118-123.

THAKOR, V.A., RAZZAQUE, M.A. & KHANDAKER, M.R.A., 2021, 'Lightweight Cryptography Algorithms for resource-constrained IoT devices: A Review, Comparison, and Research Opportunities. *IEEE Access*, 9(1):28177-28193.

TIWARI, P., SAXENA, V.P., MISHRA, R.G., & BHAVSAR, D., 2015. Wireless Sensor Networks: Introduction, Advantages, Applications, and Research Challenges. *International Journal of Technology Innovations and Research (IJTIR),* 14(1):1-11.

USMAN, M., AHMED, I., ASLAM, M.I., KHAN, S. & SHAH, U.A., 2017. SIT: A Lightweight Encryption Algorithm for Secure Internet of Things. *International Journal of Advanced Computer Science and Applications(IJACSA),* 8(1):402-411.

VASANTHI, V., 2017. Simulator and Emulator used for Wireless Sensor Network. *International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE)*, 6(1):1-10.

WEBB, J., DOCEMMILLI, F. & BONIN, M., 2015. GraphTheory Applications in Network Security in Cryptography and Security. *SSRN Electronic Journal*:1-7.

WORLU, C., JAMAL, A. & MAHIDDIN, N., 2019. Wireless Sensor Networks, Internet of Things. *International Journal of Innovative Technology and Exploring Engineering (IJITEE),* 8(12):1-12.

Appendix A: Table of Notations

Descriptions	Notations and Symbols
Cipher Key	K_C
XOR operation	$\odot$
XNOR operation	$\oplus$
Concatenation	$\|\|$
Encryption key	K_{enc}
Total energy Utilized in the network	$E_{network}$
Nonce sender	$N_{S,}$
Received message from the sender	R_S
Total energy consumption for each round	E_{round}
Cipher Text	Ct
key round	K_r
The sender of the message	S
The receiver of the message	R
Nonce receiver	N_r

Appendix B: Generation of nodes

```
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');
```

Appendix C: Graphs of results or plot

```
%%%%%%%%%%%%%%%%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
```